ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

ПРЕДВАРИТЕЛЬНЫЙ НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ПНСТ 953— 2024

СИСТЕМЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Классификация алгоритмов и вычислительных методов

(ISO/IEC TR 24372:2021, NEQ)

Издание официальное

Москва
Российский институт стандартизации
2024

Предисловие

1 РАЗРАБОТАН Федеральным государственным автономным образовательным учреждением высшего образования «Национальный исследовательский университет «Высшая школа экономики» (НИУ ВШЭ)

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 164 «Искусственный интеллект»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 8 октября 2024 г. № 56-пнст

4 Настоящий стандарт разработан с учетом основных нормативных положений международного документа ISO/IEC TR 24372:2021 «Информационная технология. Искусственный интеллект (AI). Обзор вычислительных методов для систем искусственного интеллекта» [ISO/IEC TR 24372:2021 «Information technology — Artificial intelligence (AI) — Overview of computational approaches for AI systems», NEQ]

Правила применения настоящего стандарта и проведения его мониторинга установлены в ГОСТ Р 1.16–2011 (разделы 5 и 6).

Федеральное агентство по техническому регулированию и метрологии собирает сведения о практическом применении настоящего стандарта. Данные сведения, а также замечания и предложения по содержанию стандарта можно направить не позднее чем за 4 мес до истечения срока его действия разработчику настоящего стандарта по адресу: info@tc164.ru и/или в Федеральное агентство по техническому регулированию и метрологии: 123112 Москва, Пресненская набережная, д. 10, стр. 2.

В случае отмены настоящего стандарта соответствующая информация будет опубликована в ежемесячном информационном указателе «Национальные стандарты» и также будет размещена на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет (www.rst.gov.ru)

Содержание

1 Область применения 1
2 Термины и определения 1
3 Сокращения 2
4 Общие положения 3
5 Основные характеристики систем искусственного интеллекта 5
5.1 Общие положения 5
5.2 Основные характеристики систем искусственного интеллекта 5
5.3 Вычислительные характеристики систем искусственного интеллекта 6
6 Типы вычислительных методов искусственного интеллекта 9
6.1 Общие положения 9
6.2 Методы, основанные на знаниях 10
6.3 Методы, основанные на данных 10
7 Классификация и выбор алгоритмов в системах искусственного интеллекта 11
7.1 Обзор 11
7.2 Инженерия знаний и их представления 11
7.3 Логика и рассуждения 14
7.4 Машинное обучение 16
7.5 Метаэвристика 26
Приложение А (справочное) Схема классификации алгоритмов и методов систем искусственного интеллекта 28
Библиография 30

Введение

Целью указанного стандарта является установление принципов классификации вычислительных методов и алгоритмов систем искусственного интеллекта. Внедрение данного стандарта необходимо для повышения эффективности использования систем искусственного интеллекта при решении прикладных задач.

Установление классификации вычислительных алгоритмов в соотношении с категорией систем искусственного интеллекта и их функционального назначения позволит проводить целенаправленный выбор при построении систем искусственного интеллекта.

Применение данного стандарта обеспечит повышение эффективности использования систем искусственного интеллекта для решения прикладных задач, как автономного характера, так и во взаимодействии с человеком-оператором.

Это позволит заинтересованным сторонам выбирать надлежащие решения для их приложений и сравнивать качество доступных решений.

Документ структурирован следующим образом:

- в разделе 4 приведено общее описание вычислительных подходов в системах искусственного интеллекта;
- в разделе 5 приведено основные характеристики систем искусственного интеллекта;
- в разделе 6 представлена классификация типов вычислительных подходов, включая подходы, основанные на знаниях и данных;
- в разделе 7 описаны выбранные алгоритмы, используемые в системах искусственного интеллекта, включая базовые теории и методы, их основные характеристики и типичные приложения.

ПРЕДВАРИТЕЛЬНЫЙ НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

СИСТЕМЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Классификация алгоритмов и вычислительных методов

Artificial intelligence systems.
Classification of algorithms and computational methods

Срок действия — с 2025—01—01
до 2028—01—01

1 Область применения

Настоящий стандарт содержит обзор и структурирование вычислительных подходов систем искусственного интеллекта, включая:

- основные вычислительные характеристики систем искусственного интеллекта;
- основные алгоритмы и подходы, используемые в системах искусственного интеллекта.

Стандарт предназначен для применения в сфере теоретической и практической деятельности по классификации систем искусственного интеллекта, при выполнении работ:

- по оценке соответствия вычислительных методов функциональному назначению систем искусственного интеллекта;
- по отнесению конкретных систем искусственного интеллекта к определенным классам вычислительных методов;
- по применению классификации систем искусственного интеллекта в сфере стандартизации.

В настоящем стандарте установлена схема классификации основных технологий, алгоритмов и вычислительных методов систем искусственного интеллекта для решения прикладных задач, помогающая определить направления их стандартизации. Схема классификации приведена в приложении А.

2 Термины и определения

В настоящем стандарте применены термины по [1] и [2], а также следующие термины с соответствующими определениями:

2.1

агент (agent): Физический/программный объект, который оценивает собственное состояние, состояние других объектов и окружающей среды для выполнения своих действий, включая прогнозирование и планирование, которые максимизируют успешность, в том числе при неожиданном изменении оцениваемых состояний, достижения своих целей.

[ГОСТ Р 59277—2020, пункт 3.4]

2.2 **генератор** (generator): Нейронная сеть, которая производит выборки, классифицируемые дискриминатором.

Примечание — Генераторы в основном появляются в контексте порождающих состязательных сетей.

2.3 **дискриминатор** (discriminator): Нейронная сеть, которая классифицирует выборки, создаваемые генератором.

Примечание — Дискриминаторы в основном появляются в контексте порождающих состязательных сетей.

Издание официальное

2.4

многоагентная система (multiagency system): Система, состоящая из множества взаимодействующих интеллектуальных агентов. Многоагентные системы могут решить проблемы, которые трудны или невозможны для отдельного агента или для единой (монолитной) системы.
[ГОСТ Р 59277–2020, пункт 3.31]

2.5

нечеткая логика (fuzzy logic; fuzzy-set logic): Совокупность математических теорий, основанных на принципах нечеткого множества. Нечеткая логика является одним из типов бесконечнозначной логики.
[ГОСТ Р МЭК 61131-7—2017, пункт 3.11]

2.6

онтология (ontology): Формализованное представление набора наименований понятий в рамках некоторой области, а также отношения между этими понятиями.
[Адаптировано из ГОСТ Р 56272–2014, 2.1.21]

2.7 **персептрон** (perceptron): Нейронная сеть, состоящая из одного искусственного нейрона, с двоичным или непрерывным выходным значением, которое определяется путем применения монотонной функции к линейной комбинации входных значений с обучением и исправлением ошибок.

2.8 **платформа** (platform): Комбинация операционной системы и аппаратного обеспечения, составляющая операционную среду, в которой выполняется программа.

Примечание — См. [3], статья 3.30.

2.9 **порождающая состязательная сеть;** GAN (generative adversarial network, GAN): Архитектура нейронной сети, состоящая из одного или нескольких генераторов и одного или нескольких дискриминаторов, которые конкурируют для повышения производительности модели.

3 Сокращения

В настоящем стандарте применены следующие сокращения:

ИИ (AI) — искусственный интеллект (artificial intelligence);

СИИ — система искусственного интеллекта;

ЦП (CPU) — центральный процессор (central processing unit);

AdaBoost — алгоритм машинного обучения с несколькими алгоритмами классификации (adaptive boosting);

ASIC — специализированная интегральная схема (application-specific integrated circuit);

BERT — представления двунаправленного кодера для трансформеров (bidirectional encoder representations from transformers);

BPTT — обратное распространение во времени (back propagation through time);

CNN — сверточная нейронная сеть (convolutional neural network);

DAG — ориентированный ациклический граф (directed acyclic graph);

DNN — глубокая нейронная сеть (deep neural network);

ERM — минимизация эмпирического риска (empirical risk minimization);

HMM — скрытая марковская модель (Hidden markov model);

FFNN — нейронные сети прямого распространения (feedforward neural network);

FPGA — программируемые интегральные схемы (field programmable gate array);

GDM — метод дискретизации градиента (gradient descent method);

GPU — графический процессор (graphics processing unit);

GPT — генеративный предобученный трансформер (generative pre-training);

KG — граф знаний (knowledge graph);

KNN — метод К-ближайших соседей (k-nearest neighbour)

LSTM — длинная цепь элементов краткосрочной памяти (long short-term memory)

MFCC — мелкочастотные кепстральные коэффициенты (mel-frequency cepstrum coefficient);

MLM — маскированное языковое моделирование (masked language model);

NER — распознавание именованных сущностей (named entity recognition);
NLP — обработка естественного языка (natural language processing);
NLU — система, сервис или программа понимания естественного языка (natural language understanding);
NSP — прогнозирования следующего высказывания (next sentence prediction);
OWL — язык веб-онтологий (web ontology language);
RDF — среда описания ресурса (resource description framework);
RNN — рекуррентная нейронная сеть (recurrent neural network);
RTRL — рекуррентное обучение в реальном времени (real-time recurrent learning);
SHACL — язык для описания RDF графа (shapes constraint language);
SPARQL — Протокол SPARQL и язык запросов RDF (SPARQL protocol and RDF query language);
SQL — структурированный язык запросов (structured query language);
SRM — минимизация структурных рисков (structure risk minimization);
SVM — метод опорных векторов (support vector machine);
URI — унифицированный идентификатор ресурса (uniform resource identifier);
XML — расширенный язык разметки (extensible markup language);
XLNet — модель авторегрессионных перестановок;
W3C — Консорциум Всемирной паутины (World Wide Web Consortium).

4 Общие положения

Достижения в области вычислительных подходов являются важной движущей силой в становлении искусственного интеллекта, способного выполнять различные задачи. Первоначальные методы искусственного интеллекта были в основном основаны на правилах и знаниях. В последнее время получили распространение методы, основанные на данных, такие как нейронные сети.

Вычислительные подходы с использованием искусственного интеллекта продолжают развиваться и являются важным фактором в системах искусственного интеллекта.

Вычислительные подходы для систем искусственного интеллекта делятся на категории по различным признакам. Одной из таких категорий является целевое назначение системы искусственного интеллекта. Категоризация, основанная на целях, адаптирована на основе исследований искусственного интеллекта (см. [3]). Обобщенные типы категорий включают в себя:

а) методы поиска. Эти подходы можно далее разделить на различные типы поиска: классический, алгоритмы расширенного поиска, состязательный поиск и удовлетворение ограничений:

1) классические алгоритмы поиска решают задачи путем поиска в некотором пространстве состояний и могут быть разделены на неосведомленный поиск и эвристический поиск, в которых применяется эмпирическое правило для направления и ускорения поиска;

2) расширенные алгоритмы поиска включают те, которые выполняют поиск в локальном подпространстве, являются недетерминированными, выполняют поиск с частичным наблюдением за пространством поиска, и онлайн-версии алгоритмов поиска;

3) алгоритмы состязательного поиска выполняют поиск в присутствии противника и обычно используются в играх. К ним относятся известные алгоритмы, такие как альфа-бета-обрезка, а также стохастические и частично наблюдаемые вариации Проблемы удовлетворения ограничений решаются, когда каждая переменная в задаче имеет значение, удовлетворяющее всем ограничениям;

б) логика, планирование и знания. Эти подходы можно далее разделить на три категории: логика, планирование и поиск в пространстве состояний, а также представление знаний:

1) логики, такие как пропозициональная логика и логика первого порядка, используются в классическом искусственном интеллекте для представления знаний. Решение проблемы в таких вычислительных системах включает в себя логический вывод с использованием таких алгоритмов, как разрешение;

2) планирование в классических системах искусственного интеллекта включает поиск по некоторому пространству состояний, а также алгоритмические расширения для решения задач планирования в реальном мире. Методы, позволяющие справиться со сложностью планирования в реальном мире, включают ограничения по времени и ресурсам, иерархическое планирование, при котором проблемы сначала решаются на абстрактных уровнях, а затем переходят к мелким деталям, многоагентные системы, которые справляются с неопределенностями и взаимодействуют с другими агентами в системе;

3) представление знаний — это структура данных для описания знаний с использованием логики предикатов, генерация «если — то» и представление фреймов знаний;

в) нечеткие знания и вывод. Подходы в этой области имеют дело с потенциально отсутствующими, неопределенными или неполными знаниями. Обычно они используют либо вероятность, либо нечеткую логику для представления концепций. Вероятностные вычислительные системы рассуждают, используя правило Байеса, байесовские сети или (в ситуациях, зависящих от времени) скрытые марковские модели или фильтры Калмана.

Для принятия решений используется другой набор вычислительных подходов, в том числе основанных на теории полезности и сетях принятия решений;

г) обучение. Вычислительные подходы в этой области решают проблему того, как заставить компьютер обучаться аналогично человеку. Подходы могут быть сгруппированы в обучение на примерах, обучение, основанное на знаниях, вероятностное обучение, обучение с подкреплением, подходы к глубокому обучению, GAN и другие подходы к обучению:

1) подходы к обучению, которые используются для обучения, модели на материале размеченных данных включает в себя такие методы, как деревья решений, подходы линейной и логистической регрессии, искусственные нейронные сети, непараметрические подходы (например, KNN), SVM и Ансамблевые методы обучения (например, бэггинг, бустинг);

2) подходы к обучению, основанные на знаниях, включают логику обучения, основанные на объяснениях и индуктивное логическое программирование;

3) вероятностное обучение включает в себя вычислительные подходы, такие как байесовские методы и методы максимизации математического ожидания;

4) обучение с подкреплением включает в себя вычислительные системы, которые получают обратную связь, принимают решения и предпринимают действия в окружающей среде для максимизации общего вознаграждения. Известные алгоритмы включают в себя обучение по временной разнице и Q-обучение;

5) нейронные подходы к глубокому обучению включают современные вычислительные методы со множеством скрытых уровней, включая сети с глубокой прямой связью, регуляризацию, современные методы оптимизации, CNN и методы последовательного обучения, такие как сети LSTM;

6) GAN включают в себя две конкурирующие сети, генератор и дискриминатор. Генератор генерирует образцы, а дискриминатор классифицирует каждый образец как настоящий или поддельный. После этого интерактивного процесса обученные генераторы могут быть использованы в таких приложениях, как создание искусственных изображений;

7) другие подходы к обучению включают неконтролируемое обучение, которое включает в себя определение естественной структуры наборов данных; полу-управляемое обучение, которое имеет дело с частично помеченными наборами данных; алгоритмы онлайн-обучения, которые продолжают обучение по мере получения данных; сетевое и реляционное обучение, ранжирование и обучение предпочтениям, обучение представлению, обучение передаче и активное обучение;

д) умозаключения. Эти подходы воплощают в себе применение системы искусственного интеллекта для оценки параметров или аспектов (или классификации новых или ненаблюдаемых данных) изучения, получения или определения параметров. Байесовский вывод — процесс получения статистического вывода с байесовской точки зрения. Приближенные выводы, такие как вариационный вывод, решают проблему вывода путем наилучшего приближения статистических данных. Алгоритмы Монте-Карло генерируют выборки из известного распределения, которое трудно нормализовать, а затем выводят статистику из сгенерированных выборок. Причинно-следственный вывод включает в себя установление причинно-следственных связей наблюдаемых данных;

е) уменьшение размерности. Эти вычислительные подходы предполагают уменьшение размерности данных либо с помощью алгоритмов уменьшения размерности (извлечения признаков), которые идентифицируют новое меньшее число атрибутов для представления данных, либо с помощью выбора признаков, по которым выбирается подмножество наиболее подходящих атрибутов (например, конвертация изображения из цветного в оттенки серого);

ж) общение, восприятие и действие. Вычислительные подходы в этих областях связаны с областями NLP (включая такие задачи, как языковое моделирование, классификация текстов, информационный поиск, извлечение информации, синтаксический анализ, машинный перевод и распознавание речи), компьютерное зрение (включая обработку изображений и распознавание объектов) и робототехника.

Эти категории и подкатегории не являются взаимоисключающими. Например, метод глубокого обучения [г)5)] может быть либо контролируемым [г)1)], либо неконтролируемым [г)7)], обучение с подкреплением [г)4)] может быть достигнуто с помощью глубокого обучения [г)5)] и подходов для машинного перевода или распознавания объектов [ж)], подходов к обучению [г)].

Концепции и терминология, относящиеся к вычислительным подходам искусственного интеллекта, приведены в [1].

Основа для систем искусственного интеллекта, использующих машинное обучение, включающих алгоритмы машинного обучения, алгоритмы оптимизации и методы машинного обучения, приведены в [4]. В [5] собраны и проанализированы примеры использования искусственного интеллекта.

5 Основные характеристики систем искусственного интеллекта

5.1 Общие положения

Не все системы искусственного интеллекта основаны на машинном обучении или нейронных сетях. Некоторые часто встречающиеся характеристики систем искусственного интеллекта описаны в 5.2 и 5.3. Эти характеристики носят в целом концептуальный характер и не привязаны к конкретной методологии или архитектуре. В совокупности эти характеристики отличают системы искусственного интеллекта от систем, не связанных с искусственным интеллектом. Некоторые характеристики систем искусственного интеллекта являются общими и широко применимы к различным вариантам использования, другие — для небольшого числа вариантов использования в конкретной отрасли.

Этот раздел содержит перечень характеристик систем искусственного интеллекта, который не является исчерпывающим, но содержит атрибуты, присущие многим системам искусственного интеллекта. Хотя перечень не ограничивается конкретной базовой технологией (например, системами искусственного интеллекта, построенными с использованием нейронных сетей), он не охватывает все типы динамических систем искусственного интеллекта.

5.2 Основные характеристики систем искусственного интеллекта

5.2.1 Адаптивность

Некоторые системы искусственного интеллекта адаптируются к различным изменениям во внутреннем состоянии, режимам функционирования и к среде, в которой они развертываются. Такая адаптация зависит от многих факторов, включая данные в предметной области системы, архитектуру или другие технические решения, принятые при ее внедрении. Системы искусственного интеллекта часто работают в серверных облачных вычислительных средах с доступом к высокопроизводительным вычислениям и другим ресурсам. Требования по адаптации возрастают с ростом числа систем Интернета вещей, способных выполнять вычисления общего назначения на графических процессорах и многоядерных процессорах или обрабатывать ИИ на процессорах и ускорителях для конкретных приложений. Адаптивность ИИ системы распространяется на различные аспекты внедрения Интернета вещей, такие как обработка данных практически в реальном времени, оптимизация для обеспечения низкой задержки и энергоэффективной производительности.

5.2.2 Формирование вывода

Некоторые системы искусственного интеллекта создают или генерируют статический или динамический вывод на основе заданных входных критериев. Это относится к таким методам, как обучение без учителя и генеративное обучение.

5.2.3 Координация агентов

Некоторые системы искусственного интеллекта координируют действия агентов. Агенты также могут сами по себе быть системами искусственного интеллекта, но в этом нет необходимости. Поведением агента может управлять множество одновременных ограничений, включая статические или динамические способы. Координация может проявляться либо явно посредством прямого взаимодействия между системами, либо опосредованно путем реакции на изменения в окружающей среде.

5.2.4 Динамическое принятие решений

Некоторые системы искусственного интеллекта демонстрируют динамическое принятие решений на основе внешних источников данных. Эти источники данных могут поступать с других программных платформ, из физических сред, датчиков или из других источников.

5.2.5 Возможность объяснения

Некоторые системы искусственного интеллекта предоставляют механизм для объяснения того, что ускорило принятие решения или результат. Этот вывод может принимать множество форм и может быть явным или неявным в отношении проектирования системы искусственного интеллекта. СИИ с объяснением может способствовать надежности, точности и эффективности или дополнять их. Объяснимость также может способствовать сравнению и оптимизации производительности моделей машинного обучения за счет получения информации о факторах, снижающих производительность. Объяснимость может быть важным средством для пояснения логики работы системы.

5.2.6 Дискриминативность или генеративность

Некоторые системы искусственного интеллекта являются дискриминационными, разработанными в первую очередь для того, чтобы различать возможные выходные данные, например путем исключения предшествующих вероятностей.

В качестве альтернативы некоторые системы искусственного интеллекта являются генерирующими и предназначены в первую очередь для представления соответствующих аспектов данных, например путем включения априорных вероятностей.

5.2.7 Интроспективный анализ (самодиагностика)

Некоторые системы искусственного интеллекта осуществляют самоконтроль, чтобы адаптироваться к окружающей среде или получить представление о своей функциональности, например в ситуации аудита.

Этот самоконтроль может быть адаптивным, зависящим от ситуации или статичным и может принимать различные формы в зависимости от архитектуры системы.

Для поддержки интроспективных систем искусственного интеллекта функция мониторинга производительности собирает данные и создает отчеты о показателях производительности, касающиеся вычислительных ресурсов ЦП, графического процессора или конкретного приложения, использования памяти и других системных ресурсов. Эта информация может быть использована для настройки системных ресурсов искусственного интеллекта, таких как распределение памяти, конфигурация ядра и балансировка нагрузки на многопроцессорном или гибридном компьютере. Аппаратная система позволяет системе искусственного интеллекта выполнять распараллеливание и ускорение для обучения модели машинного обучения или машинного вывода.

5.2.8 Обучение

Некоторые системы искусственного интеллекта обучаются на наборе данных перед развертыванием или обучаются динамически (путем адаптации) по мере использования системы. Системы с такими характеристиками имеют множество возможных системных архитектур (например, нейронные сети, скрытые марковские модели).

5.2.9 Идентификация и адаптация данных

Некоторые системы искусственного интеллекта имеют дело с большими объемами разнородных данных, которые являются структурированными или неструктурированными, статическими или потоковыми. Системы искусственного интеллекта могут извлекать информацию из различных наборов данных, чтобы помочь людям принимать лучшие и более точные решения.

5.3 Вычислительные характеристики систем искусственного интеллекта

5.3.1 Вычислительные характеристики, основанные на данных или знаниях

Особенностью вычислительных подходов ИИ, основанных на данных, является то, что вычислительная модель обучается на одном или нескольких источниках данных для получения знаний. Соображения, касающиеся данных, используемых в системах искусственного интеллекта, включают сбор, хранение и доступ:

- сбор данных. Вариант использования и задача приложения искусственного интеллекта обычно определяют тип данных, которые должны быть получены для обучения. Типичные задачи системы искусственного интеллекта отражены в [1], [4], включают классификацию, категоризацию (концептуальную) кластеризацию, регрессию, прогнозирование, оптимизацию, NLP (текст или речь), восприятие и системный контроль или руководство поведением. В зависимости от приложения и задачи разработчики систем искусственного интеллекта могут собирать обучающие данные с помощью интеллектуального оборудования (например, смарт-браслета, смарт-часов, смартфона), датчиков интернета вещей (например, датчика гравитации, температуры, влажности), камер, микрофонов или других датчиков;
- хранение данных. Собранные данные хранятся в формате и структуре, соответствующих ИИ приложению и задаче. Подходы к хранению и ограничения могут отличаться во время обучения и оцен-

ки. Кроме того, распределенное и совместно используемое хранилище может быть важным фактором хранения данных;

- доступ к данным. В системах искусственного интеллекта часто необходим быстрый доступ к большим объемам данных и их извлечение. Сбалансированные методы загрузки часто используются для решения проблем, связанных с параллелизмом данных и перегрузкой сети.

В дополнение к задачам персептрона и приложениям, основанным на восприятии, когнитивный интеллект стал важным аспектом СИИ, в которых когнитивные вычисления интегрированы с промышленными знаниями. Используя такие методы, как NLP и KG, системы искусственного интеллекта могут выявлять неявные знания и давать представление об отношениях, логике или паттернах, которые нелегко обнаружить наблюдателям-людям.

Пример — Используя KG, накопленные данные о бизнес-процессах могут быть преобразованы в организационный опыт и знания. Это, в свою очередь, может быть использовано для снижения затрат на коммуникацию между различными подразделениями.

5.3.2 Вычислительные характеристики на основе инфраструктуры

Системы искусственного интеллекта могут одновременно сталкиваться с проблемами оптимизации проекта вычислительной платформы, эффективности вычислений в сложных гетерогенных средах, высоко параллельных и масштабируемых вычислительных средах и вычислительной производительности приложений искусственного интеллекта. Одним из возможных решений таких проблем является использование мощных инфраструктур для обеспечения вычислительных возможностей.

Такая инфраструктура может включать в себя датчик, сервер, сеть, процессор, хранилище и другие элементы. Процессоры часто используются как для обучения, так и для аргументации в подходах к обучению. При обращении при больших объемах обучающих данных или сложных структурах DNN процессы обучения обычно требуют выполнения крупномасштабных вычислений в многопроцессорных системах или кластерах видеокарт или ASIC.

По сравнению с обучением вывод требует меньших вычислительных затрат, но все равно может потребовать значительных матричных операций, так как вывод делается для одной конкретной ситуации в информационном пространстве. Обучение и логический вывод традиционно реализовывались на облачных серверах, хотя варианты использования, требующие обработки в режиме реального времени, могут реализовывать вывод на периферийных устройствах.

В зависимости от технических архитектур процессоры на основе кремния для искусственного интеллекта включают процессоры общего назначения (например, CPU, GPU и FPGA), полукастомизированные процессоры на базе FPGA, полностью кастомизированные ASIC-процессоры и вычислительные процессоры, моделирующие работу мозга человека. Блоки обработки изображений, блоки обработки данных глубокого обучения, блоки обработки данных нейронных сетей и другие процессоры, специфичные для конкретных приложений, также подходят для различных сценариев и функций искусственного интеллекта.

Датчики с микропроцессорами для сбора, обработки и передачи информации могут быть использованы для создания полной осведомленности о внешней среде. Крупномасштабное развертывание датчиков и их применение могут поддерживать сбор данных в приложениях искусственного интеллекта.

Кроме того, для приложений «умный дом», «умная медицина» и «умная безопасность» требуются специальные требования к датчикам. Разработка интеллектуальных датчиков для приложений искусственного интеллекта основывается на таких важных факторах, как высокая точность, высокая надежность, миниатюризация и интеграция, а также высокая чувствительность.

5.3.3 Вычислительные характеристики, зависимые от алгоритма

Машинное обучение (см. [1]) есть процесс оптимизации параметров модели с помощью вычислительных методов таким образом, чтобы поведение модели отражало данные или опыт.

Методы машинного обучения находят закономерности из наблюдаемых данных или выборок и используют эти закономерности для составления прогнозов на основе входных данных без явного программирования. Методы машинного обучения частично различаются в зависимости от различий в подходах к обучению и вычислительных структур.

Метод машинного обучения включает в себя три основных элемента: функции потерь, критерии обучения и алгоритм оптимизации. Различия между методами машинного обучения можно рассматривать как функции этих элементов. Например, методы линейной классификации, такие как персептрон, логистическая регрессия и SVM, различаются с точки зрения критериев обучения и алгоритмов оптимизации.

С точки зрения функции потерь методы машинного обучения можно классифицировать как линейные или нелинейные. Надежный метод требует небольшого ожидаемого риска или ошибки, который использует функцию потерь для количественной оценки разницы между прогнозируемыми данными и реальными данными. Общие функции потерь включают функцию потерь 0-1, квадратичную функцию потерь, функцию потерь перекрестной энтропии, функцию потерь шарнира, функцию потерь средней абсолютной ошибки, функцию потерь Хубера, логарифмическую функцию потерь и функцию — квантиль потерь.

Кроме того, другие широкие категории функций потерь включают ранжирование, основанное на распределении, классификацию и регрессию.

Критерии обучения под наблюдением включают ERM и SRM. ERM уменьшает средние потери в наборе обучающих данных. SRM позволяет избежать проблем с переобучением, вводя регуляризацию параметров, основанных на ERM, для ограничения возможностей модели Обучение без присмотра имеет множество критериев обучения. Например, оценка максимального правдоподобия часто используется при оценке плотности, минимизация ошибок реконструкции — при неконтролируемом изучении признаков.

Задача оптимизации состоит в том, чтобы найти оптимальную модель машинного обучения. Она состоит из оптимизации параметров и гиперпараметрической оптимизации. Распространенные алгоритмы оптимизации включают в себя метод градиентного спуска, метод ранней остановки, пакетный градиентный спуск, стохастический градиентный спуск, мини-пакетный градиентный спуск, методы градиентного спуска, использующие коэффициент импульса, распространение среднеквадратичного значения и адаптивную оптимизацию момента. В условиях массивной обработки данных и сложной оценки знаний большая вычислительная задача обычно делится на более мелкие вычислительные задачи. Такие распределенные вычислительные платформы основаны на облачных вычислениях, пограничных вычислениях и технологиях больших данных. Платформа глубокого обучения — это базовая вычислительная платформа для глубокого обучения, которая обычно включает в себя архитектуру нейронных сетей и стабильный интерфейс глубокого обучения для поддержки распределенного обучения. Некоторые фреймворки могут быть перенесены для запуска на нескольких платформах, таких как платформы облачных вычислений и мобильные устройства.

5.3.4 Многоступенчатое или сквозное обучение

В отличие от многоступенчатого обучения, где проблема делится на несколько этапов, которые должны решаться шаг за шагом, сквозное обучение направлено на решение проблем таким образом, чтобы результаты были получены непосредственно из входных данных.

Процессы машинного обучения часто состоят из нескольких независимых модулей. Например, типичное приложение NLP включает в себя сегментацию, выделение частей речи тегами, синтаксический анализ, семантический анализ и другие независимые этапы. Каждый шаг — это индивидуальная задача, результаты каждого шага влияют на следующий шаг, потенциально влияя на весь процесс обучения.

При сквозном обучении, например при глубоком обучении, результат прогнозирования получается от входных данных к выходным. Как правило, ошибки передаются посредством обратного распространения на каждом уровне сети.

Представление каждого уровня корректируется в соответствии с такими ошибками до тех пор, пока сеть не станет конвергентной или не достигнет желаемой производительности. В таких сквозных процессах маркировка данных перед каждой самостоятельной учебной задачей больше не требуется. Если взять распознавание речи в качестве примера, то при многоступенчатом распознавании речи, как показано на рисунке 1, речь преобразуется в векторы речевых признаков (например, функции MFCC), затем группы векторов классифицируются по различным фонемам с помощью машинного обучения, исходные тексты речи с максимальной вероятностью окончательно восстанавливаются с помощью фонем. В этом процессе векторы признаков, полученные с помощью вычисления признаков, и фонемы обрабатываются акустической моделью. Акустическая модель и языковая модель обучаются отдельно.

Рисунок 1 — Распознавание речи на основе многоступенчатого обучения

Для распознавания речи на основе сквозного обучения, как показано на рисунке 2, весь процесс от выделения признаков до выражения фонемы может быть непосредственно завершен с помощью DNN. При наличии достаточного количества помеченных обучающих данных, включая пары голосовых данных и текстовых данных, в начале процесса распознавания сквозное распознавание речи, основанное на обучении, может демонстрировать хорошую производительность.

Рисунок 2 — Распознавание речи на основе сквозного обучения

6 Типы вычислительных методов искусственного интеллекта

6.1 Общие положения

Вычислительные подходы в искусственном интеллекте можно разделить на основанные на знаниях и основанные на данных.

Вычислительные подходы СИИ, основанные на знаниях, в основном состоят из ряда методов, основанных на правилах. Если взять в качестве примера экспертную систему, то обучение, рассуждение и принятие решений для конкретного варианта использования реализуются с помощью набора концептуализированных объектов и логических правил «если — то» (логического вывода). Для поддержки экспертной системы используется большая база знаний, хранящая обширные знания экспертов предметной области. В качестве структуры базы знаний используются онтологии, семантические сети, графы знаний.

Напротив, вычислительные подходы СИИ, основанные на данных, используют большие объемы данных в качестве основных ресурсов, обрабатываемых алгоритмами для моделирования человеческого мышления, обучения и процессов принятия решений. Типичные вычислительные подходы, основанные на данных, включают машинное обучение, которое может быть разложено на линейную или логистическую регрессию, вероятностную графическую модель, дерево решений, нейронные сети и другие подходы.

На рисунке 3 показана структура вычислительных подходов искусственного интеллекта.

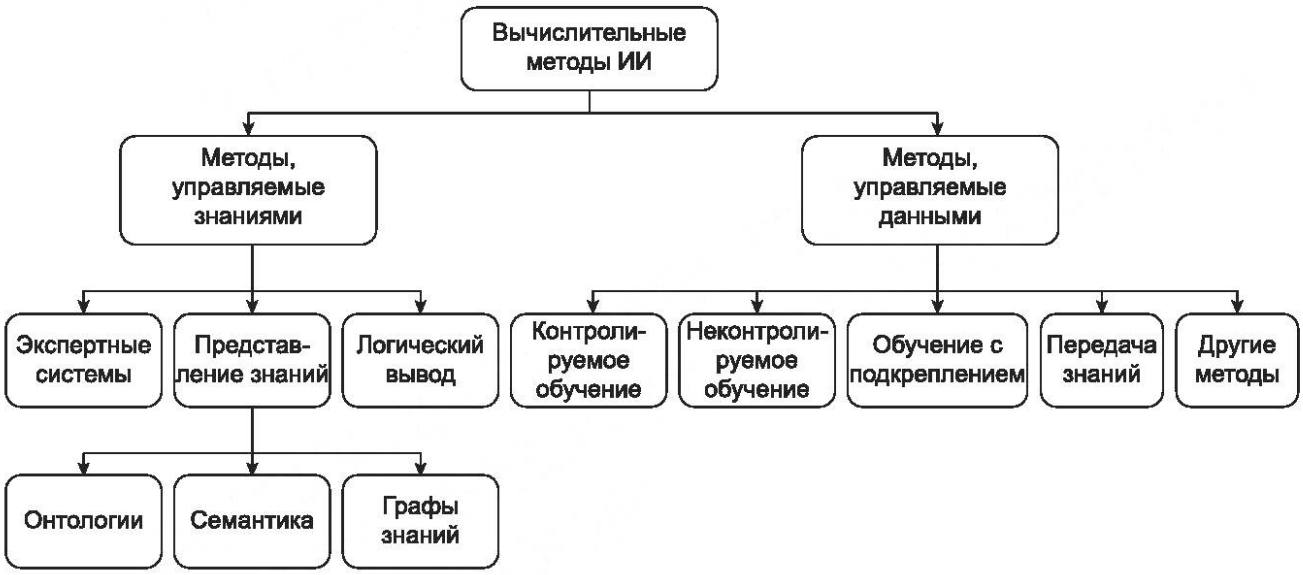


Рисунок 3 — Вычислительные подходы систем искусственного интеллекта

6.2 Методы, основанные на знаниях

Подходы, основанные на знаниях, имитируют функции человеческого интеллекта с помощью символов и логических правил. Когнитивный процесс человека рассматривается как процесс символической операции. Такой подход основан на двух основных допущениях:

- информация представлена в виде символов;
- символы обрабатываются с помощью явных правил (таких как логические операции).

6.3 Методы, основанные на данных

Подходы, основанные на данных, опираются на данные в своей вычислительной модели искусственного интеллекта. Различные подходы к машинному обучению связаны с различными типами данных, как описано в [4]. Эти подходы включают: контролируемое обучение, неконтролируемое обучение, полуконтролируемое. Ниже приведены типичные подходы к обучению.

Контролируемое обучение. Подходы к контролируемому обучению создают модели машинного обучения с использованием помеченных обучающих данных. Затем модели машинного обучения предсказывают категории или отображение входных данных. По мере того, как метки обучающих данных становятся богаче и точнее, прогнозы моделей машинного обучения и аналитика, полученная на основе этих прогнозов, становятся более полезными и надежными. Типичные алгоритмы машинного обучения для контролируемого обучения включают регрессию и классификацию. Контролируемое обучение используется в NLP, поиске информации, интеллектуальном анализе текста, распознавании рукописного ввода, обнаружении спама и других областях.

Обучение без учителя. Подходы к неконтролируемому обучению сопоставляют входные данные с выходными, используя немаркированные обучающие данные. Неконтролируемое обучение не нуждается в маркированных обучающих данных, что может снизить требования к хранению данных и вычислениям, повысить скорость работы алгоритма и помогает избежать ошибок классификации, вызванных неправильно маркированными обучающими данными. Типичные неконтролируемые алгоритмы машинного обучения (см. [2]) включают оценку плотности одного класса, уменьшение многомерности для отображения и кластеризации.

Обучение без учителя используется в экономическом прогнозировании, обнаружении аномалий, интеллектуальном анализе данных, обработке изображений, распознавании образов и других областях.

Обучение с частичным наблюдением. Подходы к обучению с частичным привлечением учителя используют как помеченные данные, так и немаркированные данные. Обучаясь на небольшом количестве помеченных данных и большом количестве немаркированных данных, такие алгоритмы могут максимально использовать ограниченные объемы помеченных обучающих данных.

Типичные алгоритмы обучения с подкреплением включают самообучение и совместное обучение.

Обучение с подкреплением использует подходы к обучению с подкреплением, которые предполагают взаимодействие агента со своей средой для достижения заранее определенной цели. Эти подходы изучают сопоставления окружающей среды с поведением, чтобы максимизировать функциональную ценность сигнала подкрепления.

В тех случаях, когда внешняя среда предоставляет мало информации, алгоритмы обучения с подкреплением полагаются на свой собственный опыт. Обучение с подкреплением успешно применяется для промышленного контроля, игры в шахматы, управления роботами, автоматизированного вождения и в других областях.

Обучение с переносом часто используется для обучения, когда в данной прикладной области отсутствуют достаточные данные для надежного обучения моделей. Трансфертное обучение с переносом — это исследовательская проблема в машинном обучении, которая фокусируется на хранении знаний, полученных при решении одной проблемы, и применении их к другой, но связанной с ней проблеме. Обучение подходит для приложений с ограниченным числом переменных, таких как классификация текста, сенсорная сеть для задач определения местоположения и классификация изображений. Машинное обучение начинается с наблюдений или получения новых данных и может быть использовано для обучающих данных, пытаясь найти закономерности, выходящие за рамки тех, которые были получены с помощью основного анализа, для реализации точных прогнозов. Алгоритмы машинного обучения включают логистическую регрессию, скрытые модели Маркова (HMM), SVM, KNN, Adaboost, байесовскую сеть, дерево решений и многие другие. Результаты подходов к машинному обучению, как правило, объяснимы с точки зрения поведения алгоритмов и моделей машинного обучения.

Машинное обучение обеспечивает основу для обучения при ограниченной доступности данных и может быть использовано для регрессионного анализа, классификации паттернов, оценки плотности вероятности и других задач. Статистика является важной теоретической основой машинного обучения, широко используемой в таких областях, как NLP, распознавание речи, изображений, поиск информации и в биологической информатике.

Нейронные сети — это сети нейронных слоев, соединенных взвешенными связями с регулируемыми весами. Нейронные сети принимают входные данные и выдают выходные данные, часто являющиеся прогнозом. Обучение нейронных сетей с несколькими скрытыми слоями называется глубоким обучением. Глубокое обучение можно рассматривать как подход к сочетанию представления функций и обучения. Глубокое обучение часто менее объяснимо, чем обычное машинное обучение. Типичные подходы к глубокому обучению включают: «deep belief network», CNN, ограниченную машину Больцмана и RNN. CNN часто используются для данных пространственного распределения, в то время как RNN часто используются для данных временного распределения, основанных на использовании ими памяти и обратной связи с предыдущими уровнями.

7 Классификация и выбор алгоритмов в системах искусственного интеллекта

7.1 Обзор

В [5] представлен шаблон, с помощью которого собирается подробная информация о вариантах использования искусственного интеллекта. Шаблон содержит описания функций вариантов использования, включая задачи, методы, платформы, топологию, используемые термины и концепции. Глубокое обучение, машинное обучение и нейронные сети являются одними из наиболее часто упоминаемых вычислительных подходов в примерах использования искусственного интеллекта.

В стандарте описывается современное состояние выбранных алгоритмов и подходов, используемых в системе искусственного интеллекта, с точки зрения теорий и методов, основных характеристик и типичных применений.

7.2 Инженерия знаний и их представления

7.2.1 Обзор

Формирование знаний, их представление и обработка — это очень широкая область в ИИ, связанная с выражением знаний в формах, поддающихся машинной обработке, и использованием машинной обработки для выполнения задач, связанных со знаниями, в частности, рассуждений во всех их формах.

Существует множество философских и практических решений, связанных с тем, как представлять знания для машинной обработки, которая должна быть одновременно точной и полезной для любых целей, которые имеют в виду люди, поэтому проектирование и инжиниринг являются важными элементами процесса. Все проекты, основанные на представлении знаний и рассуждениях, привязаны к какой-либо модели знаний, независимо от того, реализована она явно или нет.

Выбор модели знаний является ключевым фактором, определяющим, могут ли быть достигнуты цели проекта. Эта область все еще находится в стадии разработки, особенно в том, что касается технических последствий выбора.

7.2.2 Онтология

7.2.2.1 Теории и методы

Термин «онтология» происходит из философии, где он относится к тому, что люди считают, как существующее. Это имеет отношение к структурам знаний, известным как онтологии, поскольку, с философской точки зрения, оно очерчивает модель мира, которая является утверждением того, что существует. Однако «онтология» как структура знаний — это, по сути, модель знаний о любой предметной области, о которой люди хотят объявить знания. В контексте информационных технологий «онтология» обычно означает явную структуру знаний, которая имеет формальную логическую модель и поддерживает рассуждения. Наиболее распространенным видом онтологии как структуры знаний является онтология, реализуемая с помощью стека семантических веб-технологий W3C, основанного на RDF и OWL [2].

7.2.2.2 Основные характеристики

Являясь основополагающим методом когнитивного искусственного интеллекта, онтология фокусируется на концептуальной классификации сущностей и взаимосвязи между различными концепциями. Она обеспечивает формальный язык, общие определения, логическую взаимосвязь и стабильную концептуальную модель знаний, используемых в искусственном интеллекте. Онтология функционирует как схема описания знания, которая позволяет системам искусственного интеллекта приобретать, повторно использовать и обмениваться знаниями. Модель онтологии состоит из конкретно определенных концепций, которые относятся к композиционным элементам, таким как структура, сущность, термин, атрибут, функция и аксиома. Формальные описательные языки, такие как XML, RDF и OWL, используются в разработке онтологий.

Онтология — это своего рода абстрактная когнитивная модель реального мира. В рамках онтологии специально проиллюстрированы определения, атрибуты и взаимосвязи объектов. Это позволяет компьютерам и иным устройствам беспрепятственно получать и обрабатывать общие знания или знания предметной области, лежащие в основе онтологической модели.

7.2.2.3 Типичные области применения

Онтологии широко используются в инженерии знаний, например в KG, в поиске информации, взаимодействии систем и агентов, интеграции, обеспечении интероперабельности и задачах контроля качества.

7.2.3 Графы знаний

7.2.3.1 Теории и методы

Граф знаний (KG) можно рассматривать как структуру знаний, состоящую из узлов и связей. Слово «граф» относится к математическому понятию графа. Узлы на графе представляют сущности (entities), а ссылки представляют отношения между объектами. Это считается структурой знаний, потому что выражение:

entity1 -> отношение R -> entity2

принимается за утверждение, что «entity1 находится в отношениях R с entity2».

Пример — Используя Anne -> motherOf -> David утверждает, что Энн является матерью Дэвида. Эта связь является направленной, в том смысле, что Энн — мать Дэвида, но Дэвид не является матерью Энн.

KG представляют собой явный контраст с табличной структурой данных. Это более гибкий, расширяемый подход, который поддерживает более продвинутые формы рассуждения. KG не получили широкого распространения, поскольку управлять ими несколько сложнее, чем традиционными подходами.

7.2.3.2 Основные характеристики

KG представляет собой семантическую сеть, структуру данных на основе графа, состоящую из узлов и ребер. Весь граф выражает различные сущности, концепции и семантические отношения.

По сравнению с традиционными методами представления знаний (например, онтологиями или семантическими сетями), KG характеризуется:

- полнотой охвата сущностей и концепций;
- разнообразием семантических отношений;
- простотой в использовании структур графа;
- качеством представления знаний.

KG стал наиболее важным подходом к разработке знаний в СИИ, позволяющим оснастить технические системы когнитивными свойствами.

Базовый вычислительный процесс KG обычно включает в себя извлечение знаний, представление знаний, хранение знаний, моделирование знаний, объединение знаний и вычисление знаний:

а) извлечение знаний: знания извлекаются из структурированных, полуструктурированных и неструктурированных данных, особенно из текстовых данных. В соответствии с различными объектами извлечение знаний включает извлечение сущности (например, NER), извлечение отношения или свойства и извлечение события;

б) представление знаний: представление знаний — это для описания знаний с использованием логики предикатов, генерации «если — то» и фреймового представления;

в) хранение знаний: объекты хранения знаний включают базовые знания об атрибутах, знания об отношениях, знания о событиях, временные знания и знания о ресурсах. Распространенными методами хранения являются табличные и графические. В частности, хранилище знаний на основе графов включает в себя графы свойств, методы RDF и гиперграфа;

г) моделирование знаний: целью моделирования знаний является построение модели данных для KG, которая необходима для построения всего KG. Существуют методы моделирования «сверху вниз» и «снизу вверх».

Пример — Нисходящий подход определяет схему данных или онтологию, а затем постепенно уточняет ее для формирования хорошо структурированной иерархической классификации. Подход «снизу вверх» обобщает и организует существующие сущности для формирования основополагающих концепций, а затем постепенно абстрагирует их для формирования концепций более высокого уровня.

Объединение знаний обычно реализуется с уровня данных и концептуального уровня. Объединение на уровне данных фокусируется на связывании сущностей и формировании результирующих сущностей, в то время как объединение на уровне концепций фокусируется на согласовании онтологий и межъязыковом слиянии. Вычисление знаний направлено на получение неявных знаний на основе информации, предоставленной KG.

Примеры

1 Использование онтологических или основанных на правилах подходов к рассуждению об абстрактных понятиях и сущностях.

2 Использование подходов к связыванию сущностей для прогнозирования неявных связей между сущностями.

3 Использование подходов социального моделирования для обнаружения структурированных сущностей в KG и предоставления связей исследуемых объектов и знаний (правил) в KG.

7.2.3.3 Типичные области применения

KG широко используется для поиска знаний, интеллектуальных рекомендаций и задач идентификации цифровых моделей объектов, интеграции цифровых платформ.

7.2.4 Семантическая сеть

7.2.4.1 Теории и методы

Семантическая паутина — это всемирная сеть знаний. Это граф знаний, который использует стек технологий семантического веба, основанный на RDF. Все сущности и отношения в семантической сети имеют URI, поэтому они могут быть расположены в любом месте сети (однако знания, которые организации генерируют с использованием технологий семантической сети, не обязательно становятся общедоступными).

Поскольку технологии семантической сети основаны на графах, базовой единицей знаний является RDF-триплет: сущность -> связь -> сущность. URI и триплеты объединяются, образуя граф узлов URI и ссылок: RDF, RDF-schema и OWL.

Стек технологий «semantic web» относится к стандарту W3C (см. [2]) и является открытым стандартом. Стек включает в себя SPARQL — язык запросов, аналогичный SQL и SHACL — язык, основанный на представлениях форм, используемых для обеспечения соответствия графа знаний желаемым ограничениям. Как и веб, семантическая сеть децентрализована, масштабируема, расширяема и гибка. В отличие от Интернета, она предназначена для непосредственной обработки машинами, а не для взаимодействия с людьми.

7.2.4.2 Основные характеристики

Семантическая сеть основана на формальной семантике, известной как «логика описания», которая является дедуктивной формой логики. Утверждения могут иметь логические характеристики, а отношения между сущностями могут быть транзитивными или рефлексивными, что поддерживает рассуждения (например, Сара выше Энн, Энн выше Дэвида, следовательно, Сара выше Дэвида). Существуют различные средства рассуждения, доступные для выполнения функций рассуждения.

Ключевыми идеями в RDF являются классы, подклассы и экземпляры. RDF имеет два типа свойств (связей): те, которые связывают классы с другими классами, и те, которые связывают классы с литералами.

Пример — «Родитель» — это подкласс класса «человек», элемент — это подкласс класса «человек». Энн — это пример родителя. Из этого можно сделать вывод, что Энн — человек. Это форма силлогистического рассуждения, поддерживаемая наследованием классов.

Примеры синтаксисов для семантической сети включают:

- синтаксисы на основе XML;
- синтаксисы, основанные на формальной логике, которые напоминают логические формулы;

- синтаксисы, частично написанные на естественном языке (манчестерский синтаксис) для улучшения удобочитаемости для человека;
- синтаксисы, которые полностью читаемы как естественный язык, которые помогают экспертам в предметной области выполнять проверку знаний без необходимости понимать технический синтаксис.

7.2.4.3 Типичные области применения

Типичные области применения семантической сети включают разработку улучшенных возможностей поиска и поэтапное моделирование в приложениях здравоохранения.

7.3 Логика и рассуждения

7.3.1 Общие положения

Логика и рассуждения связаны с использованием существующих знаний для получения новых знаний и обеспечения того, чтобы это было сделано правильно и надежно. Формально это относится к области эпистемологии в философских исследованиях, хотя неофициально существует множество методов, основанных на здравом смысле. Это фундаментально важно для математики, естественных наук и любой сферы жизни, где важна достоверность. Искусственный интеллект открывает перспективу использования машин для автоматизации процесса, тем самым упрощая получение знаний и проверку на достоверность. Использование возможностей машин для логики и рассуждений в настоящее время находится в стадии разработки и требует большего внимания.

7.3.2 Индуктивное рассуждение

Индуктивное рассуждение связано с понятием обобщения на примерах. Оно широко используется во многих сценариях. Метод индуктивного рассуждения начинается с примеров, которые являются своего рода наблюдениями, и использует их для создания теории. При наличии достаточного количества подтверждающих доказательств некоторые теории становятся общепринятыми правилами.

Примеры

1 Ученые изучили множество форм жизни, и было обнаружено, что у всех них есть ДНК. Ученые, вероятно, обобщают из этого теорию о том, что «все формы жизни имеют ДНК». Хотя она точно описывает все формы жизни, изученные на сегодняшний день, люди не знают, верно ли это для всех форм жизни во все времена и в любом месте. Всегда остается логичной возможность того, что люди однажды изучат новую форму жизни и обнаружат, что у нее нет ДНК.

2 Бывают случаи, когда индуктивные выводы становятся фактом, но только тогда, когда это может быть доказано эмпирически. Например, основываясь на многих видах наблюдений, существовала теория, что Земля круглая. Доказательство этого заняло некоторое время, но в конце концов это было окончательно доказано эмпирическими методами и принималось как факт.

7.3.3 Дедуктивный вывод

Дедуктивное рассуждение — это форма рассуждения, которая начинается с набора пропозиций (утверждений), известных как «посылки» или «аксиомы», и использует только те методы рассуждения, которые гарантируют, что если посылки истинны, то любые сделанные выводы также истинны. Также необходимо, чтобы термины имели четкое значение. Пропозиция — это утверждение, которое может быть оценено как истинное или ложное.

Исторически дедуктивные модели мышления были известны со времен Аристотеля. Признанные формы дедуктивного рассуждения приведены в 7.3.3.1–7.3.3.5.

7.3.3.1 Пропозициональная логика

Логика высказываний или пропозициональная логика, также логика нулевого порядка — это раздел символической логики, изучающий сложные высказывания, образованные из простых, и их взаимоотношения. В отличие от логики предикатов, пропозициональная логика не рассматривает внутреннюю структуру простых высказываний, она лишь учитывает, с помощью каких союзов и в каком порядке простые высказывания сочленяются в сложные.

Она использует логические связки (и, или, не) для объединения предложений в составные формы. Есть таблицы истинности, которые помогают оценивать составные формы как истинные или ложные в зависимости от того, являются ли составные предложения истинными или ложными, например, если P истинно, а Q ложно, то (P и Q) ложно, но (P или Q) истинно, а не (P) ложно и не (Q) истинно.

Существуют более сложные формы логики, которые также используют дедуктивные методы. Они используют те же логические связи (и, или, не), что и пропозициональная логика.

7.3.3.2 Логика первого порядка

Логика первого порядка также известна как логика предикатов или логика количественной оценки. Ключевое различие между логикой первого порядка и логикой высказываний заключается в том,

что логика первого порядка использует переменные и кванторы. Используемые кванторы — «все» и «существует». В то время как в пропозициональной логике делаются только такие утверждения, как «Сократ — личность», в логике первого порядка можно сказать: «существует x такой, что x — Сократ, а x — личность». Это позволяет использовать новые дедуктивные формы рассуждения. Существуют также высшие формы логики, которые добавляют выразительности, но теряют способность рассуждать.

7.3.3.3 Модальная логика — это раздел формальной логики, который занимается изучением логических операторов — модальностей.

В качестве стандартных рассматривают модальности: возможно, необходимо. В современной логике модальными операторами считают большинство операторов, служащих для учета степени истинности утверждаемого.

В рамках модальной логики выделяют подразделы, например:

- временную логику;
- логику знаний;
- динамическую логику;
- логику доказуемости.

7.3.3.4 Пространственная логика: если A находится перед B, а B — перед C, то A находится перед C. «Перед» понимается как переходное отношение. Подразумевается точка зрения наблюдателя, поскольку понятие «фронт» относительно.

7.3.3.5 Временная логика: если A происходит после B, а B происходит после C, то A происходит после C. Здесь A, B и C — события, и предполагается, что «происходит после» является переходной зависимостью.

7.3.4 Гипотетические рассуждения

Гипотетические рассуждения широко используются в науке и математике, но могут быть применены в любом аспекте жизни. Они используется в криминалистике, юридических спорах и повседневных рассуждениях.

В качестве отправной точки они принимают утверждение, которое потенциально является истинным или ложным, но никто не знает, каким именно. Это известно как «гипотеза», и в науке и математике обычно это научная или математическая теория, которую люди хотят проверить.

Из гипотезы логически вытекают следствия относительно рассматриваемой сущности или явления. В идеале должна существовать возможность их эмпирической проверки. В идеале эта сущность (или явление) может быть проверена эмпирически, но во многих случаях это не так, и тогда говорят, что теория «не поддается проверке», т. е. нельзя провести никаких тестов, которые показали бы, что она ложна.

Используемой формой рассуждения является:

гипотеза: P

предпосылка: если P, то Q.

Теперь вывод зависит от того, окажется ли Q истинным. Если это ложно, то P является ложным, и гипотеза P была фальсифицирована. Это дедуктивный вывод: гарантируется, что P равно неправда. Если Q оказывается истинным, это не доказывает, что P истинно, это просто согласуется с тем, что P истинно. Затем нужно выдвинуть другую предпосылку и другое условие для проверки.

Пример — Гипотеза: сегодня утром шел дождь. Предпосылка: если сегодня утром шел дождь, дорожка будет мокрой. Эмпирическое наблюдение: влажна ли дорожка? Эмпирическим результатом может быть «да» или «нет» (это может быть представлено в виде ветви дерева решений).

Эмпирический результат 1: нет; вывод: сегодня утром дождя не было (если предположение верно и наблюдение достоверно, это дедуктивно верно).

Эмпирический результат 2: да; заключение: нет.

То, что тропинка мокрая, логически согласуется с тем, что сегодня утром шел дождь, но это ничего не доказывает. Возможно, тропинка мокрая из-за того, что ее поливали из шланга или по другой причине. Всегда необходимо «предположение об открытом мире фактов», пока не будут получены дополнительные факты.

В повседневной жизни люди часто принимают истинность Q за доказательство истинности P, что является логической ошибкой, известной как «утверждение следствия»[1)].

[1)] Диагностические рассуждения: врачи часто берут группу симптомов, чтобы указать на наличие определенного состояния, и ставят диагноз, утверждая, что «симптомы X, Y и Z соответствуют состоянию C». Технически это «подтверждение следствия»: диагност предполагает, если другого возможного объяснения нет, что касается симптомов. См. 7.3.5.

В математике схема принятия гипотезы, допущения ее истинности и приведения к ложному выводу называется «доказательством от противного». Это показывает, что гипотеза неверна, показывая, что гипотеза и вывод логически не согласуются друг с другом.

7.3.5 Байесовский вывод

Байесовский вывод — статистический вывод, в котором свидетельство и/или наблюдение используются, чтобы обновить или вновь вывести вероятность того, что гипотеза может быть верной.

Алгоритм Байеса — это статистический метод, который используется для определения вероятности событий на основе предыдущих знаний об этом событии. Этот метод основан на теории вероятности, которая позволяет нам оценить вероятность случайного события, на основе его значимости и частоты его возникновения.

Наивный байесовский алгоритм — это алгоритм классификации, основанный на теореме Байеса с допущением о независимости признаков. Другими словами, алгоритм предполагает, что наличие какого-либо признака в классе не связано с наличием какого-либо другого признака.

Байесовский метод — это способ формализации степени разумной уверенности в некотором утверждении, и ее корректировки по мере поступления информации относительно исследуемого явления.

Байесовский вывод — форма статистического вывода. Следует отметить, что статистический вывод является особым подходом по сравнению с математическим выводом. Чистая математика касается только дедуктивного вывода: если математическое доказательство достоверно, то выводы всегда логически вытекают из посылок[1)].

7.4 Машинное обучение

7.4.1 Общие положения

Подходы к машинному обучению имеют большое значение для практики и научных разработок. Алгоритмы машинного обучения упоминаются в большинстве примеров использования стандарта [5].

В этом разделе представлены такие методы обучения, как дерево решений, случайный лес, линейная или логистическая регрессия, KNN, наивный байесовский подход и подходы, связанные с нейронными сетями.

7.4.2 Дерево решений

7.4.2.1 Теории и методы

Деревья решений представляют контролируемые алгоритмы машинного обучения, которые рекурсивно разбивают пространство данных на основе значений атрибутов. Секционирование описывается как DAG, где узлы в графе связаны с тестами атрибутов, а конечные узлы соответствуют либо значению класса, либо значению регрессии. На рисунке 4 показан пример классификации набора данных растения. Правила можно считывать из дерева решений. Например, «Если ширина лепестка <= 0,8, то класс = setosa» или «Если ширина лепестка > 0,8 и длина лепестка <= 4,75, то класс = versicolor».

Внутренние узлы могут иметь два или более дочерних узлов. Узлы, связанные с числовыми атрибутами, часто разделяются на два дочерних узла с помощью бинарного теста, как показано на рисунке 4. Узлы, связанные с категориальными атрибутами, либо используют двоичный тест (т. е. атрибут равен или не равен определенному значению), либо, что более распространено, для каждого возможного значения, которое принимает атрибут, есть дочерний элемент.

[1)] Судить о том, является ли математическое доказательство достоверным, — интересное упражнение, поскольку оно зависит от способностей человека к математическому мышлению, но вообще говоря, если доказательство достоверно, то все математики соответствующей квалификации могут согласиться, что это так. В некоторых областях, таких как теория чисел, некоторые гипотезы содержат проверяемые прогнозы о поведении чисел, но в большинстве областей чистой математики нет ничего, что можно было бы эмпирически проверить в качестве перекрестной проверки доказательства. См. 7.3.4.

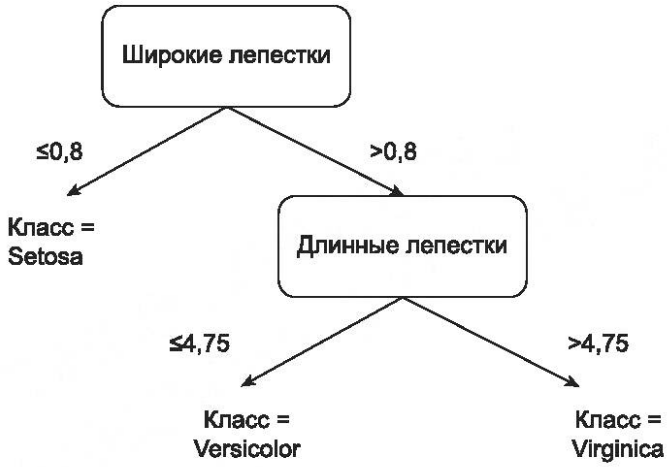


Рисунок 4 — Пример дерева решений

7.4.2.2 Основные характеристики

Алгоритмы индукции дерева решений строят дерево решений для определенного набора обучающих данных. Это включает в себя использование статистического теста для выбора наилучшего атрибута для разделения данных на каждом шаге. Тест разбивает на разделы данные из родительского узла на основе выбранного атрибута. Желаемыми тестами являются те, которые уменьшают количество классов в данных, связанных с родительским узлом, до наборов данных, содержащих в основном тот же класс (т. е. менее неупорядоченный или более чистый), связанный с дочерними узлами. Распространенными статистическими тестами являются прирост информации или индекс Джини. Это приводит к семейству алгоритмов, связанных с деревом классификации и деревом регрессии (см. [2]).

Ввод дерева решений продолжается до тех пор, пока не будут достигнуты некоторые критерии завершения. Общие критерии включают остановку, когда все данные на узле имеют одинаковое значение класса или когда с узлом связано меньше указанных точек данных. Максимальное построение дерева до тех пор, пока все данные в узлах не будут принадлежать одному классу, обычно приводит к переобучению. Следовательно, индукция дерева либо завершается до того, как это произойдет (известно как ранняя остановка), либо максимальное дерево будет обрезано после завершения обучения.

Индукция дерева решений представляет собой простой алгоритм машинного обучения с контролем, создающий деревья, которые относительно легко понять человеку. Поскольку индукция дерева решений не требует значительной предварительной обработки данных, она обычно выполняется быстро. Алгоритм обрабатывает числовые и категориальные данные, а также пропущенные значения. Однако из-за своей относительной простоты он часто не дает таких точных результатов, как другие контролируемые методы.

7.4.2.3 Типичные области применения

Индукция дерева решений может генерировать четкие древовидные структуры для различных результатов прогнозирования на основе признаков. Обычно это объяснимо экспертам в предметной области и может быть использовано для таких задач, как классификация и прогнозирование.

7.4.3 Метод случайного леса

7.4.3.1 Теории и методы

Случайный лес (см. [2]) — метод машинного обучения с коллективным управлением. Он состоит из набора из n деревьев решений, каждое из которых построено для решения задачи машинного обучения. Большинство голосов деревьев принятия решений в ансамбле используется для прогнозирования или классификации новых точек данных.

Дерево решений t_i в ансамбле строится из начальной загрузки выборки b_i обучающего набора. Образец начальной загрузки — это образец, взятый с заменой из обучающего набора. В среднем он содержит около двух третей точек данных из выборочного набора данных. Это означает, что n образцов начальной загрузки немного отличаются друг от друга, что приводит к разнообразию деревьев в ансамбле, что в свою очередь способствует надежности обучаемого ансамбля в целом.

Случайные леса также вносят разнообразие в ансамбль деревьев, ограничивая выбор атрибутов, подлежащих тестированию при построении дерева, которое должно состоять из m случайно выбранных атрибутов из набора доступных атрибутов. Обычно m намного меньше, чем d, общее количество

атрибутов. Часто значение m устанавливается равным $\sqrt{d}$. Таким образом, основными параметрами, управляющими алгоритмом случайного леса, являются n и m, а также тип построенного дерева решений.

7.4.3.2 Основные характеристики

Случайные леса обладают двумя полезными свойствами: можно вычислить тестовую ошибку для ансамбля, не прибегая к набору отклонений, и можно оценить относительную важность атрибутов в наборе данных для задачи машинного обучения.

Эти свойства обусловлены использованием образцов, взятых из упаковки. Выходной набор o_i для *i-го* дерева решений — это набор точек данных из обучающего набора, которые не выбраны из начальной загрузки b_i. В связи с тем, что o_i не используется для обучения дерева решений t_i, он используется для оценки его тестовой ошибки. Ошибка тестирования всего ансамбля может быть оценена.

Аналогично, путем перестановки значений определенного атрибута «а» и вычисления разницы в ошибке теста для дерева до и после перестановки можно оценить важность атрибута. Большие различия в оценочной ошибке теста указывают на то, что атрибут «а» важен для решения проблемы. И наоборот, небольшая разница в оценочной ошибке теста или ее отсутствие указывают на то, что атрибут «а» неважен. Путем усреднения по нескольким перестановкам и деревьям алгоритм вычисляет среднее уменьшение ошибки, которое затем используется для ранжирования важности атрибутов. Иногда вместо точности используется среднее значение снижения индекса Джини.

Как метод ансамбля, случайный лес обладает наибольшими преимуществами при интерпретации (особенно с использованием оценки переменной важности) и несмещенной тестовой ошибкой для ансамбля. Однако иногда он плохо работает с наборами данных с огромным количеством частично коррелированных атрибутов.

7.4.3.3 Типичные области применения

Метод случайного леса — это надежный контролируемый алгоритм машинного обучения. Интерпретации результирующего моделирования способствует переменная оценка важности отдельных деревьев в лесу. Переменная важность деревьев из случайного леса также обычно используется в качестве метода выбора признаков для предварительной обработки данных.

7.4.4 Линейная регрессия

7.4.4.1 Теории и методы

Линейная регрессия — это метод регрессии, который строит функцию независимых переменных для прогнозирования значений некоторых целевых переменных.

Линейная регрессия — это самый простой тип регрессионного анализа. В линейной регрессии ожидаемое значение y, заданное набором независимых переменных $x = \{x_1, x_2, ..., x_p\}$, где p — число независимых переменных, показано в формуле (1):

$$y = b_0 + b_1x_1 + b_2x_2 + \ldots + b_px_p, \quad (1)$$

где b_0, b_1, b_2 *и* b_p *— коэффициенты модели.*

Как только коэффициенты b_p оценены, можно предсказать значение y, учитывая новые значения независимой переменной x.

В линейной регрессии цель состоит в том, чтобы найти линейную гиперплоскость, которая наилучшим образом соответствует точкам данных. Наиболее распространенным методом поиска параметров является обычная регрессия методом наименьших квадратов, где наилучшим соответствием является гиперплоскость, которая минимизирует квадрат ортогонального расстояния между точками данных и гиперплоскостью. Такие показатели, как значения R^2- и остаточная стандартная ошибка, часто используются для оценки соответствия модели, а F-статистика используется для оценки значимости линейной зависимости.

7.4.4.2 Основные характеристики

Модели линейной регрессии оперируют несколькими допущениями. Во-первых, предполагается, что взаимосвязь между зависимыми и независимыми переменными является линейной. Во-вторых, ошибки (т. е. разница между фактическими и расчетными значениями y) независимы. В-третьих, дисперсия ошибок постоянна по отношению к ответу. В-четвертых, предполагается, что ошибки происходят из нормального распределения. Наконец, предполагается, что независимая переменная x измерена без ошибок. Модель линейной регрессии может быть легко расширена для создания полиномиальных и других нелинейных регрессионных моделей. Чтобы распространить действие на модель полиномиальной регрессии, члены в модели линейной регрессии заменяются полиномиальными членами, такими как x^2. Можно также расширить линейную регрессию, чтобы учесть взаимодействия между независи-

мыми переменными путем добавления членов, которые являются результатом умножения независимых переменных.

7.4.4.3 Типичные области применения

Линейная регрессия используется, когда существует линейная зависимость между независимыми и зависимыми переменными, и необходимо найти значение зависимой переменной. Например, при прогнозировании стоимости дома с учетом цен на жилье и особенностей местности.

7.4.5 Логистическая регрессия

7.4.5.1 Теории и методы

Логистическая регрессия — это метод классификации, при котором взаимосвязь между независимой переменной X и зависимой переменной Y моделируется в соответствии с формулой (2):

$$\text{Log}\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p\, p, \qquad (2)$$

где $p(X) = P(Y = 1|X)$;

β_0, β_1, β_2, β_p — коэффициенты.

Левая часть формулы представляет собой логарифмическое преобразование коэффициентов $p(X)$, и логистическая регрессия предполагает, что оно может быть выражено в виде линейной функции. Коэффициенты логистической регрессии обычно оцениваются с использованием метода максимального правдоподобия. Максимальное правдоподобие направлено на оценку параметров таким образом, чтобы прогнозируемая вероятность $p(X)$ каждой выборки была как можно ближе к фактическому значению выборки.

7.4.5.2 Основные характеристики

Логистическая регрессия вычисляет вероятность того, что данное исходное значение принадлежит к определенному классу. Она используется для задач классификации: оценивает апостериорные вероятности принадлежности данного объекта к тому или иному классу. Для оценки модели логистическая регрессия применяет метод максимального правдоподобия.

7.4.5.3 Типичные области применения

Логистическая регрессия используется для небольших наборов данных, где существует линейная зависимость между независимыми и зависимыми переменными. Логистическая регрессия больше всего подходит для наборов данных, в которых есть только два класса. Для множественной классификации более подходят такие методы как линейный дискриминантный анализ.

7.4.6 Метод «K-ближайший сосед»

7.4.6.1 Теории и методы

KNN — это метод машинного обучения, который используется как для классификации, так и для регрессии. Его иногда называют «ленивым» алгоритмом, поскольку он не извлекает функцию из обучающих данных. А KNN является непараметрическим методом, поскольку он не предполагает фиксированной структуры модели.

KNN делает прогноз для новой точки данных, находя «ближайших соседей» в обучающих данных. Ближайшие соседи определяются с помощью показателей расстояния. Общие показатели включают расстояния: Евклидово (Euclidean), Манхэттен (Manhattan) и Махаланобис (Mahalanobis). При классификации алгоритм находит K-ближайших соседей, и метка класса является наиболее распространенной меткой в этих ближайших соседях. При регрессии оценочное значение часто является средним значением для ближайших соседей.

7.4.6.2 Основные характеристики

Поскольку KNN зависит от мер расстояния, обычно рекомендуется нормализовать значения непрерывных атрибутов, чтобы определенные атрибуты не превышали результирующую меру расстояния. Поскольку KNN является непараметрическим методом, он особенно подходит для наборов данных, в которых отсутствуют четкие границы принятия решений или которые нелегко смоделировать.

Одним из недостатков KNN является то, что не существует удовлетворительного способа работы с категориальными атрибутами.

7.4.6.3 Типичные области применения

KNN — простая для понимания и реализации модель, и она часто дает приемлемые результаты. Следовательно, это хороший базовый метод, который стоит попробовать перед более продвинутыми методами. Поскольку для выполнения классификации или регрессии KNN требует использовать весь набор данных, прогнозирование может быть очень медленным при большом размере данных или высокой размерности данных.

7.4.7 Наивный байесовский подход

7.4.7.1 Теории и методы

Наивный байесовский классификатор (Naive Bayes classifier) — вероятностный классификатор на основе формулы Байеса со строгим (наивным) предположением о независимости признаков между собой при заданном классе, что сильно упрощает задачу классификации из-за оценки одномерных вероятностных плотностей вместо одной многомерной.

В данном случае, одномерная вероятностная плотность — это оценка вероятности каждого признака отдельно при условии их независимости, а многомерная — оценка вероятности комбинации всех признаков, что вытекает из случая их зависимости. Именно по этой причине данный классификатор называется наивным, поскольку позволяет сильно упростить вычисления и повысить эффективность алгоритма.

Наивный байесовский подход обобщается в некое представление или модель. Наивная байесовская модель использует байесовскую теорию для обучения и вывода, как показано в формуле (3).

Сама же формула Байеса выглядит следующим образом:

$$P(A|B) = \{P(B|A)\ P(A)\}/\{P(B)\}, \quad (3)$$

где $P(A|B)$ — апостериорная вероятность события A при условии выполнения события B;

$P(B|A)$ — условная вероятность события B при условии выполнения события A;

$P(A)$ и $P(B)$ — априорные вероятности событий A и B соответственно.

7.4.7.2 Основные характеристики

Чтобы найти вероятности для дискретной переменной, значения каждого возможного класса или значения атрибута сводятся в таблицу. Если переменная непрерывна, то необходимо оценить функцию распределения вероятностей переменной.

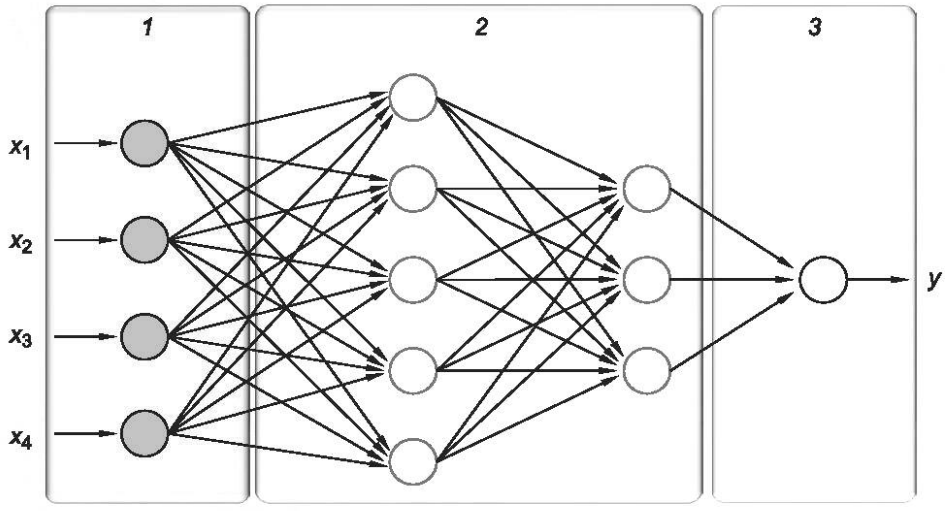
7.4.7.3 Типичные области применения

Для классификации используются наивные байесовские модели, преимущественно при небольшом объеме набора данных. Такой набор данных может быть использован для многоклассовой классификации. Его не рекомендуется применять, если не выполняется предположение об условной независимости атрибутов. Для небольших наборов данных, если $P(Ai|C)$ равно нулю, все предсказание не выполняется.

7.4.8 Нейронная сеть прямого действия

7.4.8.1 Теории и методы

Нейронные сети FFNN имеют топологическую структуру, в которой каждый нейрон принадлежит к определенному слою. Каждый слой нейронов получает входные сигналы от нейронов предыдущего слоя и выходные сигналы к нейронам следующего слоя. Слои между входным слоем и выходным слоем являются скрытыми слоями. Сигнал распространяется от входного уровня к выходному слою в одном направлении без обратной связи, которая может быть представлена DAG. На рисунке 5 показан пример FFNN



1 — входной слой; *2* — скрытый слой; *3* — выходной слой

Рисунок 5 — Пример структуры FFNN

7.4.8.2 Основные характеристики

FFNN обладают высокой способностью к подгонке и могут аппроксимировать общие непрерывные нелинейные функции, которые могут быть использованы для преобразования сложных признаков или аппроксимации сложного условного распределения.

В машинном обучении входные характеристики оказывают большое влияние на классификатор. Если взять в качестве примера контролируемое обучение, то эффективные классификаторы могут значительно повысить производительность. Следовательно, для достижения приемлемых результатов классификации необходимо определение признака, при котором вектор выборки преобразуется в более эффективный вектор признаков. Поскольку многослойный FFNN можно рассматривать как нелинейную составную функцию, он также может быть применен в качестве метода преобразования признаков, где его выходные данные могут использоваться в качестве входных данных классификатора для классификации. Когда структура скрытых слоев и нейронов оптимизирована, многослойные FFNN могут точно аппроксимировать сложные непрерывные функции. Одним из слабых мест FFNN является то, что они подвержены переобучению, так что модель не способна надежно обобщаться на новые данные.

7.4.9 Рекуррентная нейронная сеть

7.4.9.1 Теории и методы

RNN — это тип нейронной сети, обладающей возможностями кратковременной памяти. Нейроны могут получать информацию как от других нейронов, так и от самих себя в RNN, образуя, таким образом, сетевую структуру с петлей. На рисунке 6 показан пример RNN. Математически RNN можно рассматривать как динамическую систему, которая использует функцию для описания изменения всех состояний в заданном пространстве со временем. Полностью подключенная RNN аналогична любой нелинейной динамической системе.

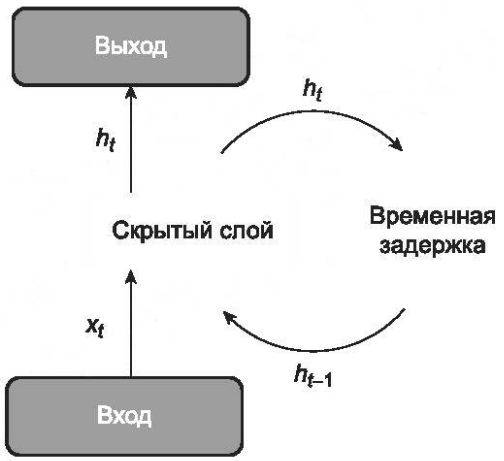


Рисунок 6 — Пример структуры RNN

7.4.9.2 Основные характеристики

RNN могут применяться к различным типам задач машинного обучения, включая следующие задачи:

- задача преобразования последовательности в категорию в основном используется для классификации данных последовательности, где модель машинного обучения принимает данные последовательности в качестве входных данных и выводит данные категории. Например, при классификации текста входными данными является последовательность слов, выходными — категория текста;
- задача синхронного преобразования последовательности в последовательность в основном используется для маркировки последовательностей, где ввод и вывод применяются к каждому образцу, длина входной и выходной последовательностей одинакова.

Например, при тегировании частей речи каждое слово должно быть помечено соответствующим тегом части речи;

- в асинхронных задачах «последовательность за последовательностью», также известных как модель «кодер — декодер», входные и выходные последовательности не обязательно должны строго соответствовать друг другу и не обязательно поддерживать одинаковую длину. Например, при машин-

ном переводе входными данными является последовательность слов исходного языка, выходными — последовательность слов целевого языка.

GDM часто используется для изучения параметров в RNN. Для вычисления градиента используются методы BPTT и RTRL. Функция BPTT заключается в передаче информации об ошибке вперед шаг за шагом в соответствии с обратным порядком времени. Поскольку размерность вывода RNN ниже чем размерность ввода, BPTT обладает характеристиками меньшего объема вычислений, но большей пространственной сложности. Поскольку метод RTRL не нуждается в градиентной обратной связи, он подходит для задач, требующих онлайн-обучения. Относительно длинная входная последовательность вызовет проблему градиентного взрыва и исчезновения, также известную как проблема долгосрочной зависимости. Для смягчения этой проблемы были разработаны усовершенствования RNN, такие как стробирующие механизмы.

7.4.9.3 Типичные области применения

RNN широко используются в распознавании речи, языковом моделировании и генерации естественного языка.

7.4.10 Сеть долгой краткосрочной памяти (LSTM)

7.4.10.1 Теории и методы

Сеть LSTM — это специальная RNN, которая может получать информацию о долгосрочной зависимости. Входные элементы, элементы забывания и выходные элементы используются в сети LSTM для управления информационным потоком. Входные элементы выборочно «запоминаются». Элемент забывания выборочно забывает входные данные с предыдущего узла. Выходной элемент определяет, какие выходные данные будут обрабатываться как текущее состояние.

7.4.10.2 Основные характеристики

Сеть LSTM может изучать зависимости на больших расстояниях благодаря внедрению механизма, который является наиболее важной характеристикой для контроля циркуляции и потери функций. Скорость отзыва по сети LSTM высока, поскольку существует сильная зависимость от дальности действия и объем данных, которые могут быть обработаны, велик. Сеть LSTM может использоваться как своего рода «промежуточное состояние» для описания последовательности, и результаты также могут быть применены в качестве функций для последующего использования.

Сети LSTM устраняют проблему увеличения градиента или исчезновения градиента простым способом RNN. Сети LSTM в настоящее время могут обрабатывать последовательности величиной 100 порядков, но проблема исчезающего градиента все еще может сохраняться для последовательностей величиной 1000 порядков или дольше.

Поскольку каждая ячейка сети LSTM имеет четыре полностью подключенных уровня, если временной интервал сети LSTM велик и сеть глубока, это вычисление может быть большим и отнимать много времени.

Сети LSTM также теоретически способны подбирать произвольные функции, в которых предположения и ограничения, связанные с проблемой, значительно ослаблены.

7.4.10.3 Типичные области применения

Приложения сети LSTM включают машинный перевод, языковое моделирование и распознавание речи.

7.4.11 Сверточная нейронная сеть

7.4.11.1 Теории и методы

CNN — это нейронная сеть с прямой связью. CNN — это многослойные персептроны, отражающие биологический образ мышления. CNN включают в себя входной слой и комбинацию слоев свертки, слоев активации, объединяющего слоя, полностью подключенного слоя и слоев нормализации. Пример структуры CNN в виде модели машинного обучения показан на рисунке 7.

7.4.11.2 Основные характеристики

CNN — это, по сути, отображение ввода-вывода. Она может изучить большое количество отображающих соотношений между входными данными и выходными данными без какого-либо точного математического выражения между входными данными и выходными данными. CNN обучена на обучающей выборке, до тех пор, пока данные, используемые для работы, релевантны обучающей выборке, сеть имеет возможность корректно предсказывать (выходные данные).

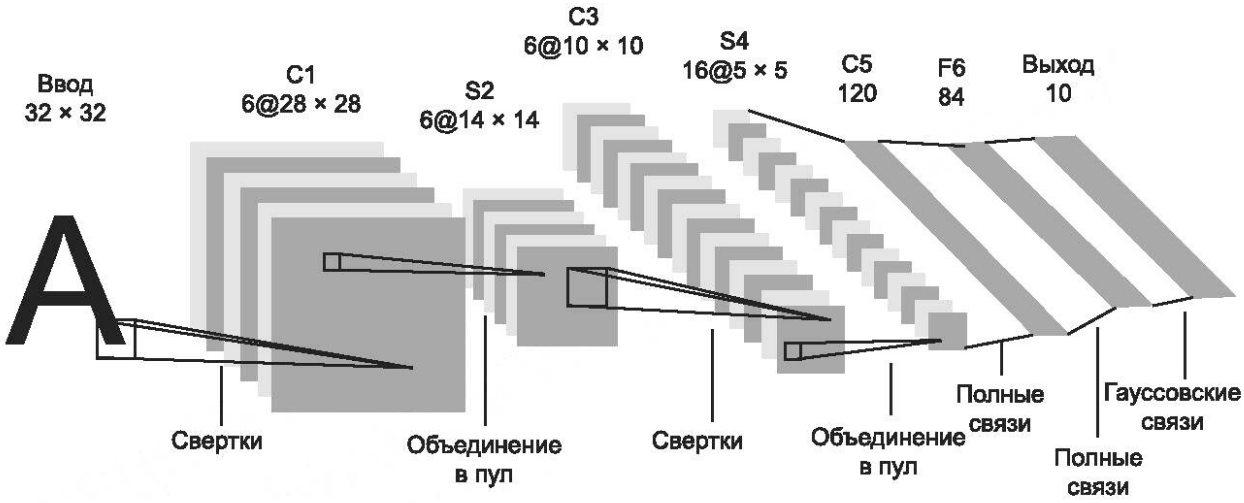


Рисунок 7 — Пример структуры модели машинного обучения CNN: сеть LeNet5

Одной из наиболее важных характеристик CNN является то, что она имеет форму перевернутого треугольника, что ограничивает чрезмерную потерю градиента в нейронных сетях обратного распространения.

При использовании машинного обучения для обработки изображений ядро свертки перемещается по изображению (или определенному объекту), чтобы получить новый набор функций с помощью операций свертки. Слой свертки используется для извлечения локальных объектов, где ядра свертки функционируют как средства извлечения объектов. CNN обладают двумя важными особенностями:

- локальные соединения, где каждый нейрон в сверточном слое подключен только к нейрону в определенном локальном окне на следующем слое, образуя локальную сеть соединений;
- распределение веса, при котором фильтры для всех нейронов на слое одинаковы.

В этом случае сеть может обучаться параллельно, что также является одним из преимуществ CNN по сравнению с нейронными взаимосвязанными сетями.

Поскольку CNN совместно используют ядро свертки, потребность в обработке многомерных данных снижается. Для оптимизации производительности (например, для точного прогнозирования классов изображений) могут потребоваться значительные ручные усилия при выборе функций, тренировке веса и настройке параметров.

Часто требуются большие наборы обучающих данных, что в свою очередь может потребовать использования графических процессоров, многоядерных процессоров или процессоров для конкретных приложений. CNN также не всегда поддаются объяснению, поскольку неясно, какие аспекты входных данных отражаются в выходных отображениях.

7.4.11.3 Типичные области применения

CNN широко используются для распознавания изображений и классификации, например в приложениях для распознавания лиц.

7.4.12 Порождающая состязательная сеть

7.4.12.1 Теории и методы

GAN используют состязательное обучение и генеративные модели для получения выборок, соответствующих реальным распределениям данных. GAN обучают сети-генераторы, которые производят выборки, предназначенные для того, чтобы вызвать неправильную классификацию со стороны сети-дискриминатора.

GAN также обучают сети-дискриминаторы, которые пытаются определить, является ли выборка реальными данными или создается сетью-генератором.

Таким образом, две противоположные сети постоянно обучаются. Конвергенция происходит, когда сеть-дискриминатор больше не может точно классифицировать выборки как настоящие или поддельные, так что сеть-генератор выдает выборки, соответствующие реальному распределению данных.

7.4.12.2 Основные характеристики

GAN могут представлять скрытые измерения и демонстрировать скрытые взаимосвязи между данными. По сравнению с задачей оптимизации с одной целью, цели оптимизации сети-генератора и сети-дискриминатора в GAN противоположны. Поэтому процесс обучения GAN может быть сложным

и нестабильным, возможности двух сетей должны быть сбалансированы. Если сеть дискриминатора слишком надежна на начальных стадиях состязательности, то сеть генератора затем не сможет улучшиться. В процессе обучения требуется оптимизация гиперпараметров таким образом, чтобы на каждой итерации сеть дискриминатора была немного более надежной, чем сеть генератора.

7.4.12.3 Типичные области применения

GAN обычно используются в задачах обработки изображений, таких как преобразование текста в изображение, перевод изображения в изображение и рисование изображений.

7.4.13 Трансферное обучение

7.4.13.1 Теории и методы

Методы передачи знаний хранят и абстрагируют знания, полученные из обучающих данных для данной задачи, и применяют эти знания к другой задаче. Знания в форме модели машинного обучения при некоторых обстоятельствах могут быть адаптированы к новой задаче или предметной области. Это особенно полезно, когда трудно или невозможно получить и пометить обучающие данные для этой новой задачи или предметной области

7.4.13.2 Основные характеристики

Трансферное обучение основано на выявлении возможностей применения существующих знаний в новой предметной области, передаче помеченных данных или структур знаний (например, моделей машинного обучения) для использования в этой новой предметной области и настройке или оптимизации передаваемых знаний применительно к новой целевой области или задаче.

Трансферное обучение предназначено для уменьшения зависимости от больших объемов маркированных данных, относящихся к конкретной предметной области. Однако данные не всегда поддаются адаптации к новым областям, и передаваемые знания не всегда адекватно отражают характеристики данных в новых доменах.

В зависимости от того, что передается, подходы к трансферному обучению могут основываться на передаче объектов, передаче функций или совместном использовании параметров. Трансферное обучение также может быть классифицировано как индуктивное или трансдуктивное. Как правило, при индуктивном обучении с переносом исходные и целевые домены связаны друг с другом, и исходная область опирается на большое количество обучающих выборок. При обучении трансдуктивному переносу процесс передачи происходит с использованием данных из разных доменов.

7.4.13.3 Типичные области применения

Типичные приложения для обучения передаче данных включают распознавание изображений и NLP, где модели, обученные переводу текста между двумя языками, используются для перевода текста на третий язык. Аналогичный подход используется при взаимодействии онтологий.

7.4.14 Двунаправленные представления кодировщика для трансформеров

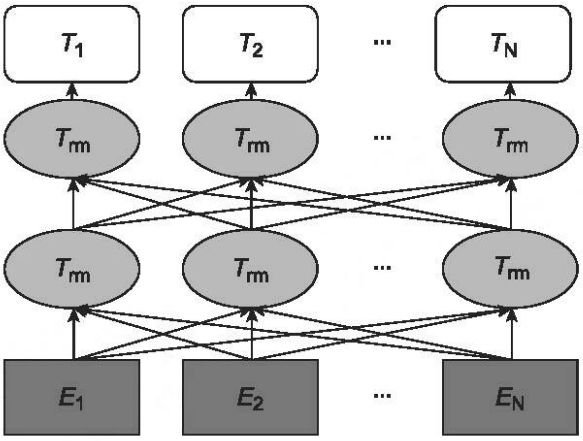
7.4.14.1 Теории и методы

BERT — это своего рода языковая модель, которая использует немаркированные обучающие данные для получения обширного семантического представления текста. Это представление, усвоенное в ходе выполнения конкретной задачи NLP, в свою очередь может быть повторно использовано для решения других задач NLP, часто после тонкой настройки или путем применения последующих алгоритмов. BERT представляет собой многослойный двунаправленный трансформатор-кодировщик. Его входные данные включают в себя ряд комбинированных векторов, таких как символы, текст и информация о местоположении. Его результатом является семантический вектор.

Типичная структура для модели BERT показана на рисунке 8.

7.4.14.2 Основные характеристики

BERT предназначен для двунаправленной обработки слова с использованием информации. В отличие от предварительной подготовки к другим языковым моделям, методы BERT не предсказывают наиболее вероятное слово на основе всех предыдущих слов. Вместо этого он случайным образом маскирует некоторые слова и использует все открытые слова для составления прогноза. BERT устраняет ограничение однонаправленности, используя предварительное обучение MLM. Получив предложение, MLM случайным образом стирает одно или несколько слов и пытается предсказать стертые слова на основе оставшихся слов. Это делает модель более зависимой от контекстуальной информации для прогнозирования и дает модели определенную возможность исправления ошибок. BERT также использует NSP, чтобы определить, следует ли второе предложение за первым в тексте.



E_1, E_2 и E_N — встроенные представления; T_1, T_2 и T_N — конечные выходы; T_{rm} — промежуточные представления данного токена

Рисунок 8 — Пример структуры модели BERT

Выполняя совместное обучение с задачами MLP и NSP, модель машинного обучения выводит векторное представление каждого слова, которое всесторонне и точно описывает общую информацию входного текста (одно предложение или пара предложений). Последующие задачи тонкой настройки обеспечивают лучшие начальные значения параметров модели машинного обучения. Однако любые несоответствия в процессах предварительной подготовки и генерации снижают эффективность в задачах генерации естественного языка.

У BERT есть этапы предварительной подготовки и тонкой настройки. Алгоритм обучается на немаркированных данных по различным задачам в ходе предварительного обучения. Затем параметры алгоритма точно настраиваются с использованием помеченных данных из конкретных задач. Типичные параметры тонкой настройки включают размер пакета, скорость обучения и количество эпохи. BERT тренирует глубокое двунаправленное представление путем совместной настройки двунаправленных преобразователей на всех уровнях. Следовательно, требуется только дополнительный уровень вывода для точной настройки предварительно обученной модели машинного обучения для различных задач, и для ряда задач нет необходимости разрабатывать архитектуру, специфичную для конкретной задачи. Чрезмерное использование масок при обучении может повлиять на производительность модели машинного обучения труднопрогнозируемым образом.

7.4.14.3 Типичные области применения

BERT внес свой вклад в развитие задач NLP, таких как корпус лингвистической приемлемости (CoLA)., Microsoft research paraphrase corpus (MRPC), многожанровый вывод на естественном языке (MultiNLI), вывод на естественном языке вопросов (QNLI), пары вопросов quora (QQP), распознавание текстового вывода (RTE), тест семантического сходства текстов (STS-B) и Stanford sentiment treebank (SST-2).

7.4.15 Авторегрессионная языковая модель XLNet

7.4.15.1 Теории и методы

XLNet представляет собой модель языка авторегрессионных перестановок и является дальнейшим усовершенствованием GPT и BERT.

XLNet непрерывно предсказывает следующее слово слева направо, не в естественном порядке предложений, а в порядке предсказания. Реализован механизм саморегулирования с двойным потоком. Этот «двойной поток» представлен в виде двух преобразователей для каждого уровня модели, потока запросов и контекстного потока.

7.4.15.2 Основные характеристики

Чтобы решить проблему двунаправленного контекста, модель языка перестановок использует информацию о местоположении цели при составлении прогнозов. Поток содержимого кодирует весь контент на текущий момент, в то время как поток запросов ссылается на предыдущую историю и текущее положение, подлежащее прогнозированию. XLNet пытается получить двунаправленную контекстную информацию, максимизируя логарифмическую вероятность всех возможных прогнозов и факта.

XLNet является авторегрессионным, в отличие от BERT. Это свойство обеспечивает большую согласованность в том, как данные отображаются в модели между этапами предварительного обучения

и точной настройки. Это свойство также позволяет избежать предположения BERT о независимости признаков.

Реализация XLNet часто требует больших вычислительных затрат.

7.4.15.3 Типичные области применения

Типичные области применения XLNet включают понимание прочитанного, ранжирование документов, задачи контроля качества, классификацию текста и задачи обработки естественного языка, связанные с пониманием смысла высказываний (Natural Language Understanding, NLU).

7.5 Метаэвристика

7.5.1 Общие положения

Метаэвристика относится к классу алгоритмов оптимизации, которые обеспечивают приемлемые решения для задачи оптимизации. Как правило, они не определяют оптимальное решение, но могут быстро найти разумное решение с использованием эвристики или правила для поиска в пространстве решений. Обычно они не делают строгих предположений о решаемой задаче, что означает, что их можно использовать для решения широкого круга задач, особенно тех, которые являются стохастическими или недифференцируемыми. Однако, поскольку они не делают строгих предположений о проблеме, они обычно работают хуже, чем алгоритмы, разработанные для конкретных задач. Метаэвристика часто основана на популяции, проводя поиск по многим точкам в пространстве поиска в то же время, это означает, что они, как правило, избегают локальных оптимумов. Они также часто являются стохастическими алгоритмами. Обычно используемые метаэвристические алгоритмы включают алгоритмы эволюционных вычислений, типичные для генетических алгоритмов, генетического программирования или эволюционных стратегий, имитационного отжига и алгоритмов роевого интеллекта.

7.5.2 Генетические алгоритмы

7.5.2.1 Теории и методы

Генетические алгоритмы — одна из самых ранних и наиболее известных форм эволюционных вычислений. Они действуют на основе генетики и отбора наиболее приспособленных геномов.

Существуют различные теории, лежащие в основе генетических алгоритмов. Наиболее распространенной является гипотеза строительного блока, которая показывает, как генетические алгоритмы собирают воедино короткие, хорошо приспособленные строительные блоки решения для создания более производительных решений.

7.5.2.2 Основные характеристики

Решение задачи оптимизации с использованием генетического алгоритма предполагает возможность представления потенциальных решений проблемы в геноме и возможность оценки генома с помощью функции приспособленности, которая оценивает геном в зависимости от того, насколько хорошо он решает проблему. Геном обычно состоит из простого списка битов, двойников или других символов, часто представляющих параметры в задаче оптимизации. Геномы также могут быть более сложными представлениями, такими как деревья, матрицы или строки, которые необходимо интерпретировать с помощью простого компьютерного языка.

Алгоритм состоит из четырех этапов: инициализация, отбор, генетические операторы и завершение. Алгоритм порождает последовательные поколения геномов, причем последующие поколения лучше решают проблему.

На этапе инициализации создается популяция геномов. Как правило, используются случайно сгенерированные геномы, но можно засеять популяцию слабыми решениями. Однако важно, чтобы в популяции существовало разнообразие исходных геномов.

После инициализации выполняется выбор. Здесь геномы в популяции оцениваются с учетом функции пригодности, и случайным образом отбираются для разведения, с вероятностью отбора, пропорциональной пригодности. Другими словами, геномы с высокой степенью соответствия, которые лучше решают проблему, с большей вероятностью будут отобраны для селекции. Распространенные подходы к выбору включают в себя выбор колеса рулетки и выбор метода ранжирования.

Как только пары геномов отобраны для селекции, применяются генетические операторы. Существует целый ряд генетических операторов, наиболее распространенными из которых являются кроссовер (или рекомбинация) и мутация.

Кроссовер объединяет части двух родительских геномов для получения потомков. Распространенными вариантами кроссовера являются одноточечный кроссовер или двухточечный кроссовер. По сути, одна или две точки разреза случайным образом устанавливаются по всей длине генома. Дочер-

ний геном содержит первую часть одного родительского генома до точки отсечения и вторую часть другого родительского генома после точки отсечения. Для двухточечного варианта, первая и третья части генома происходят от одного родителя, а вторая часть — от другого родителя. Цель кроссовера состоит в том, чтобы поделиться некоторыми полезными решениями для выявления более сильных детей. Более слабые дети удаляются в последующих поколениях путем отбора.

Другим основным генетическим оператором является мутация, при которой символы в геноме изменяются случайным образом. Цель мутации состоит в том, чтобы избежать преждевременной конвергенции путем повторного внесения разнообразия в популяцию. Кроссовер применяется с высокой вероятностью, обычно около 0,6–0,9, а мутация — с очень низкой частотой, обычно 0,1 или меньше. Последовательные поколения популяции создаются таким образом до тех пор, пока алгоритм не завершится с использованием критериев завершения, которые могут быть такими же простыми, как запуск в течение заранее определенного числа поколений, или до тех пор, пока лучший член популяции не будет удовлетворять некоторым предопределенным критериям.

7.5.2.3 Типичные области применения

Генетические алгоритмы используются в самых разнообразных приложениях. Они успешны там, где задачи не дифференцируемы, где функция пригодности не может быть записана математически и где проблемный домен является стохастическим.

Приложение А
(справочное)

Схема классификации алгоритмов и методов систем искусственного интеллекта

Таблица А.1

Алгоритмы и методы	Область применения
А. Инженерия знаний и их представления	
Онтологии	Инженерия знаний, построение графов знаний, поиск информации, задачи контроля качества
Графы знаний	Поиск знаний, хранение знаний, взаимодействие с онтологиями, выработка интеллектуальных рекомендаций и задач контроля качества
Семантическая сеть	Разработка улучшенных возможностей поиска и поэтапное моделирование в приложениях здравоохранения
Б. Логика и рассуждения	
Индуктивное рассуждение	Идентификация знаний
Дедуктивный вывод	Логика: пропозиционная; логика первого порядка; модальная логика; пространственная логика; временная логика
Гипотетические рассуждения	Оценка гипотез и вывод
Байесовский вывод	Проверка гипотез, диагностические решения в медицине и технике
В. Машинное обучение	
Дерево принятия решений	Индукция дерева решений может генерировать четкие древовидные структуры для различных результатов прогнозирования на основе признаков. Обычно это объяснимо экспертам в предметной области и может быть использовано для таких задач, как классификация и прогнозирование
Случайный лес	Надежный контролируемый алгоритм машинного обучения. Качеству интерпретации результирующего моделирования способствует переменная оценка важности и осмотр отдельных деревьев в лесу. Переменная важность из случайного леса также обычно используется в качестве метода выбора признаков для предварительной обработки данных
Линейная регрессия	Регрессия используется, когда существует линейная зависимость между независимыми и зависимыми переменными, и необходимо найти значение зависимой переменной. Используется для прогнозирования
Логическая регрессия	Регрессия используется для небольших наборов данных, где существует линейная зависимость между независимыми и зависимыми переменными. Логистическая регрессия больше всего подходит для наборов данных, в которых есть только два класса
К-ближайшие соседи	Базовый метод, который стоит применять перед более совершенными методами, поскольку он требует, чтобы весь набор данных для выполнения классификации или регрессии, при прогнозировании должен быть очень медленно меняющимся

Окончание таблицы А.1

Алгоритмы и методы	Область применения
Байесовский метод	Для классификации используются наивные байесовские модели, особенно когда набор данных не слишком велик. Он может быть использован для классификации при множестве классов. Но не рекомендован, если не выполняется предположение об условной независимости атрибутов и для очень небольших наборов данных
Нейронная сеть прямого действия	Многослойные FFNN могут точно аппроксимировать сложные непрерывные функции. Слабым местом FFNN является то, что они подвержены переобучению, так что модель не способна надежно обобщаться на новые данные
Рекуррентная нейросеть	RNN широко используются в распознавании речи, языковом моделировании и генерации естественного языка
Сеть долгой краткосрочной памяти	Приложения сети LSTM включают машинный перевод, языковое моделирование и распознавание речи
Сверточная нейронная сеть	CNN широко используются для распознавания изображений и классификации, например в приложениях для распознавания лиц
Трансферное обучение	Типичные приложения для обучения передаче данных включают распознавание изображений и NLP, где модели, обученные переводу текста между двумя языками, используются для перевода текста на третий язык
Порождающая состязательная сеть	Типичные приложения для обучения передаче данных включают распознавание изображений и NLP, где модели, обученные переводу текста между двумя языками, используются для перевода текста на третий язык
Двунаправленное представление кодирования для трансформеров	BERT вносит свой вклад в развитие задач НЛП, таких как корпус лингвистической приемлемости (CoLA)., Microsoft research paraphrase corpus (MRPC), многожанровый вывод на естественном языке (MultiNLI), вывод на естественном языке вопросов (QNLI), пары вопросов quora (QQP), распознавание текстового вывода (RTE), тест семантического сходства текстов (STS-B) и набор данных Stanford sentiment treebank (SST-2)
Авторегрессионная языковая модель XLNet	Типичные области применения XLNet включают понимание прочитанного, ранжирование документов, задачи контроля качества, классификацию текста и NLU
Г. Метаэвристика	
Генетический алгоритм	Генетические операторы, генетический отбор, генетические алгоритмы используются в самых разнообразных приложениях. Они эффективны для решения недифференцируемых задач, в которых функция пригодности не может быть записана математически, и где проблемная область является стохастической

Библиография

[1] ИСО/МЭК 22989:2022 Информационная технология. Искусственный интеллект. Концепции и терминология искусственного интеллекта (Information technology — Artificial intelligence — Artificial intelligence concepts and terminology)

[2] ISO/IEC TR 24372:2021 Информационная технология. Искусственный интеллект (AI). Обзор вычислительных методов для систем искусственного интеллекта [ISO/IEC TR 24372:2021 Information technology — Artificial intelligence (AI) — Overview of computational approaches for AI systems, NEQ]

[3] ISO/IEC/IEEE 26513:2017 Системная и программная инженерия. Требования для тестирования и оценки информации пользователем (Systems and software engineering — Requirements for testers and reviewers of information for users)

[4] ИСО/МЭК 23053:2022 Структуры для систем искусственного интеллекта, использующие машинное обучение [Framework for Artificial Intelligence (AI) Systems Using Machine Learning (ML)]

[5] ISO/IEC TR 24030:2024 Информационные технологии. Искусственный интеллект. Примеры применения [Information technology — Artificial Intelligence (AI) — Use cases]

УДК 004.01:006.354 ОКС 35.020

Ключевые слова: системы, искусственный интеллект, классификация, алгоритмы, вычислительные методы

Технический редактор *И.Е. Черепкова*
Корректор *О.В. Лазарева*
Компьютерная верстка *И.А. Налейкиной*

Сдано в набор 10.10.2024. Подписано в печать 23.10.2024. Формат 60×84⅛. Гарнитура Ариал.
Усл. печ. л. 4,18. Уч.-изд. л. 3,55.

Подготовлено на основе электронной версии, предоставленной разработчиком стандарта

Создано в единичном исполнении в ФГБУ «Институт стандартизации»
для комплектования Федерального информационного фонда стандартов,
117418 Москва, Нахимовский пр-т, д. 31, к. 2.
www.gostinfo.ru info@gostinfo.ru